



**MINUTES**  
**OF**  
**SIXTH MEETING OF UG BOARD OF STUDIES (BoS)**  
**OF**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**NIT ANDHRA PRADESH**



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH  
संगणक विज्ञान एवं अभियांत्रिकी विभाग  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**Panel of UG Board of Studies (BoS)  
for  
Department of Computer Science and Engineering, NIT Andhra Pradesh**

S. No.	Name	Designation and Affiliation	Role
1.	Dr. Karthick S	Assistant Professor DCSE	Member
2.	Dr. K. Himabindu	Assistant Professor, DCSE	Member
3.	Dr. S. Nagesh Bhattu	Assistant Professor & Head DCSE	Chairman and Secretary
4.	Dr. Srilatha Chebrolu	Assistant Professor, DCSE	Member
5.	Dr. Choudhary Shyam Prakash	Assistant Professor, DCSE	Member
6.	Prof. R.B.V. Subramanyam	Professor, NIT Warangal, DCSE	External Academic Expert
7.	Prof. Mary Saira Bhanu	Professor, NIT Trichy, DCSE	External Academic Expert
8.	Dr. Sridhar Chimalakonda	Associate Professor, DCSE, IIT Tirupati	External Academic Expert
9.	Dr. Kollaparthi Venkateswara Rao	Senior Systems Architect, Virtusa	External Industry Expert
10	Dr CH.N.A.B.Sankar	Scientist 'G', RCI, DRDO, Hyderabad	External Industry Expert
11.	Mr. Abhijat Ande	Software Engineer II, Akamai, Bangalore	Alumni
12.	Dr. N. Jayaram	Dean Academic Affairs	Invitee
13.	Dr. R. Sunil Kumar	Associate Dean (Academics)	Invitee



### Minutes of Third Meeting of UG Board of Studies

The Board of Studies (BoS) of the Department of Computer Science and Engineering, NIT Andhra Pradesh was held on 09.04.2025 from 2:00 PM onwards in the online mode through GMeet. The following members were present at the Third Meeting of the UG Board of Studies of the Department of Computer Science and Engineering.

S. No.	Name	Designation and Affiliation	Role
1.	Dr. Karthick S	Assistant Professor, DCSE	Member
2.	Dr. K. Himabindu	Assistant Professor, DCSE	Member
3.	Dr. S. Nagesh Bhattu	Assistant Professor & Head DCSE	Chairman and Secretary
4.	Dr. Srilatha Ch.	Assistant Professor, DCSE	Member
5.	Dr. Choudhary Shyam Prakash	Assistant Professor, DCSE	Member
6.	Prof. R.B.V. Subramanyam	Professor, NIT Warangal, DCSE	External Academic Expert
7.	Prof. Mary Saira Bhanu	Professor, NIT Trichy, DCSE	External Academic Expert
8.	Dr. Sridhar Chimalakonda	Associate Professor, DCSE, IIT Tirupati	External Academic Expert
9.	Dr. Kollaparthi Venkateswara Rao	Senior Systems Architect, Virtusa	External Industry Expert
10.	Dr CH.N.A.B.Sankar	Scientist 'G', RCI, DRDO, Hyderabad	External Industry Expert
11.	Mr. Abhijat Ande	Software Engineer II, Akamai, Bangalore	Alumni
12.	Dr. N. Jayaram	Dean Academic Affairs	Invitee
13.	Dr. R. Sunil Kumar	Associate Dean (Academics)	Invitee

The Chairman, BoS welcomed the members of the BoS. The Chairman invited the Course designers to present the syllabi of the II-year CSE courses.



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH  
संगणक विज्ञान एवं अभियांत्रिकी विभाग  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

---

Further, the agenda points of the meeting have been taken up sequentially as mentioned in the Table below:

Item No.	Item Description	Page No.
1.	To consider and approve revision of ESC(Engineering Science Courses) and OPC (Open Elective Courses) courses offered by DCSE	5
2.	To consider and approve the Syllabi of core courses of the III-year and IV B.Tech to be offered to the B.Tech. (CSE) students under R24-Revision of the Curriculum	8-10
3.	To consider and approve the Syllabi of elective courses of the III-year and IV B.Tech courses to be offered to the B.Tech. (CSE) students under R24-Revision of Curriculum	11-14
4.	To consider and approve the Syllabi of the minor and double major degree programmes offered by the CSE Department.	14



**ITEM NO. 1**

**Sub:** To consider and approve revision of ESC (Engineering Science Courses) and OPC (Open Elective Courses) courses offered by DCSE and renaming of the course titled Language Processors

The course (CS2011) Modelling and Optimization Techniques is currently offered as ESC course for CSE department (offered by CSE Department). As per the guideline for R24 curriculum revision (as per the resolution of Senate item S23.5.1.c) the number of ESC credits that can be offered by a Department for its B.Tech. programme is upper capped to a total of 6 credits. As per the current curriculum structure, the number of ESC credits offered by CSE department to B.Tech. CSE programme amounts to a total of 8 credits. After careful analysis of the current curriculum, it is proposed to split the ESC course Modelling and Optimization Techniques (CS2011) into two courses Discrete Event Simulation (CS2011) of 1 Credit and Modelling and Optimization Techniques (EC2511) of 2 credits. The modified course codes and credit distribution is summarized below.

S. No.	Course Code	Course Title	Credit Structure
1.	CS2011	Discrete Event Simulation	1-0-0
2.	EC2511	Modelling and Optimization Techniques	2-0-0

The current contents of the Modelling and Optimization Techniques (CS2011) was carefully looked at by the BoS members. As the current course has a natural structure for enabling the splitting of the course into two courses which can be taught separately, it was suggested to split the course in the proposed structure. Detailed contents of the revised content of these courses is provided in the Annexure – B

Currently the programming courses are delivered to all the departments as part of I B.Tech. curriculum. A theory and lab course combination of CS1011 and CS1012 with titles



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH  
संगणक विज्ञान एवं अभियांत्रिकी विभाग  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Problem Solving through Computer Programming & Problem Solving through Computer Programming Lab are part of current curriculum.

However, after careful analysis of the contents of the course, the heads of respective departments requested for the delivery and level of the course that is suitable for their department students as many a times a mismatch between the content and level required for their department students is resulting in large number of failures and also depriving the students from important placement opportunities.

Considering the representations from the department specific content proposals, it is proposed to deliver the programming courses in three different modes with suitable modification in the titles.

The course offered to CSE students is proposed with title “Introduction to Computational Problem Solving (CS1011)” with associated lab titled “Introduction to Computational Problem-Solving Lab (CS1012)” and is to be delivered keeping the rigor required for CSE students.

The course offered to ECE & EEE students is proposed with title “Problem Solving through Computer Programming (CS1031)” with associated lab titled “Problem Solving through Computer Programming Lab (CS1032)” and is to be delivered with rigor kept at the level of subsequent data structures courses in I-II.

The course offered to the engineering branches of Biotechnology, Chemical, Civil, Mechanical, Materials and Metallurgical are proposed with titles “Introduction to Computer Programming (CS1041)” with associated lab titled “Introduction to Computer Programming Lab (CS1042)” and is to be delivered at the level of introducing a programming course required for engineering branches.

Course Code	Course Title	L	T	P	Credits	Cat	Offered to Dept
CS1011	Introduction to Computational Problem-Solving	3	-	-	3	ESC	CSE
CS1012	Introduction to Computational Problem-Solving Lab	-	1	2	2	ESC	CSE
CS1031	Problem Solving through Computer Programming	3	-	-	3	ESC	ECE & EEE



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH  
संगणक विज्ञान एवं अभियांत्रिकी विभाग  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CS1032	Problem Solving through Computer Programming Lab	-	1	2	2	ESC	ECE & EEE
CS1041	Introduction to Computer Programming	3	-	-	3	ESC	Biotech, Chemical, Civil, Mechanical, MME
CS1042	Introduction to Computer Programming Lab	-	1	2	2	ESC	Biotech, Chemical, Civil, Mechanical, MME

The Syllabi presented is attached in Appendix-A of this minute's document.

The department of ECE has requested for offering a one credit course titled "Basics of Algorithm Design" under ESC category with the contents of course presented in Annexure-B. This proposed course covers for required ESC credits offered by other departments to the ECE department and is meant for improving employability of ECE students. All the ESC courses offered by CSE are summarily presented in Annexure B.

The Open Elective courses offered by DCSE are presented in Annexure J. These courses are already approved in previous BOS and are presented here for completeness.

It is proposed to rename the following existing courses with a revised title to comply with standard naming used for these courses.

Course Code	Existing course Name	Revised course name
CS2071	Language Processor	Compiler Design
CS2072	Language Processor Lab	Compiler Design Lab

**Resolution:**

The BoS members approved the proposed changes as the justification provided by the Chairman was perceived to be sound. The approved courses include Annexure B, Annexure D and Annexure J and overall curriculum which is part of Annexure A.



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH  
संगणक विज्ञान एवं अभियांत्रिकी विभाग  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**ITEM NO. 2**

**Sub:** To consider and approve the syllabi of core courses of the III-year and IV B.Tech to be offered to the B.Tech. (CSE) students under R24-Revision of the Curriculum. The overall curriculum structure of III and IV B.Tech. are as depicted in the following table.

*Table 1III and IV B.Tech. Overall Curriculum Structure*

III – Year: I – Semester							
S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS3011	Machine Learning	3	-	-	3	PCC
2	CS3012	Machine Learning Lab	-	-	2	1	PCC
3	CS3021	Software Engineering	2	-	-	2	PCC
4	CS3022	Software Engineering Lab	-	-	2	1	PCC
5	CS3031	Computer Networks	2	1	-	3	PCC
6	CS3032	Computer Networks Lab	-	-	2	1	PCC
7		Department Electives				6	DEC
8		Open Elective /DAC approved Free Electives (NPTEL, MOOCs, etc.)	3	-	-	3	OEC
9	SM3021	Introduction to Design Thinking				1	HSC
		Total				21	
III – Year: II – Semester							
S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS3051	Distributed Computing	3	-	-	3	PCC
2	CS3052	Distributed Computing Lab	-	-	2	1	PCC
3	CS3061	Cryptography and Network Security	3	-	-	3	PCC
4	CS3071	Theory and Design of Programming Languages	2	1	-	3	PCC
5		Department Electives				6	DEC
6		Open Elective /DAC approved Free Electives (NPTEL, MOOCs, etc.)	3	-	-	3	OEC
7	HS3011	English for Engineers – II(LCA-I)	2	1	-	3	HSC
8	SM3011	Introduction to Entrepreneurship				1	HSC



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH  
संगणक विज्ञान एवं अभियांत्रिकी विभाग  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

		Total					23	
<b>IV – Year: I – Semester</b>								
S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code	
1		Open Elective /DAC approved Free Electives (NPTEL, MOOCs, etc.)	6	-	-	6	OEC	
2	CS4014	Professional Major Work	-	-	12	6	PRC	
3		Department Electives				5	DEC	
4		Liberal Arts/Creative Arts Courses – II				3	HSC	
		Total				20		
<b>IV – Year: II – Semester</b>								
S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code	
1		Semester-Long Internship (SLI) /Additional Project at the institute/Additional department elective courses for 6 credits	-	-	12	6	SLI	
		Total				6		

The BoS members reviewed the curriculum structure and suggested minor modifications for core courses as enlisted below

1. In general, the course content has to be presented uniformly across all the courses
2. The lab course content can be presented to the level of topics covered rather than detailed experiment list
3. Software Engineering course content should be revised with the inclusion of generative tools
4. Some of the topics related to semi-supervised learning may be removed to make it fit for in-depth coverage of other topics and better feasibility of coverage in 3 credit structure.
5. It was suggested to offer a basic course on cryptography through a traditional course with title “Cryptography and Network Security” rather than the proposed “Cryptography and Engineering Secure Systems”.
6. Computer Networks Course should include topics covering IPv6
7. Software Engineering Course needs slight revision in the light of recent trends in the usage of AI tools
8. A pre-requisite graph must be included in the course structure to better make the course dependencies easily understood.



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH  
संगणक विज्ञान एवं अभियांत्रिकी विभाग  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Core Courses of III B.Tech. I Sem (Offered by CSE)							
SL. No	Course Code	Course Title	L	T	P	Credits	Course Designer
1	CS3011	Machine Learning	3	-	-	3	SNB
2	CS3012	Machine Learning Lab	-	-	2	1	SNB
3	CS3021	Software Engineering	2	-	-	2	SNB
4	CS3022	Software Engineering Lab	-	-	2	1	SNB
5	CS3031	Computer Networks	3	-	-	3	CSP
6	CS3032	Computer Networks Lab	-	-	2	1	CSP
Core Courses of III B.Tech. II Sem (Offered by CSE)							
6	CS3051	Distributed Computing	2	1	-	3	SNB
7	CS3052	Distributed Computing Lab	-	-	2	1	SNB
8	CS3061	Cryptography and Network Security	3	-	-	3	SNB
9	CS3071	Theory and Design of Programming Languages	2	1	-	3	KS

The Syllabi pertaining to revision are presented are attached in Annexures as summarized in the table

S.NO	Annexure	Content Description	Comment
1	Annexure-C	I-B.Tech. Core Courses	It is the same content as presented in the previous BOS. There is no revision for course content for any of the courses, except for formatting
2	Annexure-D	II-B.Tech. Core Courses	It is the same content as presented in the previous BOS. There is no revision for course content for any of the courses, except for formatting.
3	Annexure-E	III-B.Tech. Core Courses	It contains the course content of III B.Tech. core courses revised as suggested in BOS meeting



**Resolution: The BOS members approved the revised curriculum contents of the courses which are presented in the Annexures C, D, E**

**ITEM NO. 4**

**Sub:** To consider and approve the Syllabi of the III-year and IV B.Tech. elective courses to be offered to the B.Tech. (CSE) students under R24-Revision of Curriculum.

The members presented the detailed contents of elective courses which are part of curriculum structure of III and IV B.Tech. are as depicted in the following table.

**Sem-IV Elective Courses**

SL. No	Course Code	Sub Name	Credits	CAT	Course Designer
1	CS2402	Data Processing and Visualization Lab	3	DS	SNB
2	CS2412	Modelling and Optimization Lab	3	TCS	SNB
3	CS2422	Mobile Application Development Lab	3	CSA	SNB
4	CS2432	Software Coding Standards Lab	3	CSA	KHB
<b>Sem-V Elective Courses</b>					
1	CS3401	Data Warehousing and Data Mining	3	DS	CHS
2	CS3411	Data Science	3	DS	CHS
3	CS3421	Advanced Database Systems	3	SY	CHS
4	CS3431	Graph Algorithms	3	TCS	KS
5	CS3441	Randomized Algorithms	3	TCS	KS
6	CS3451	Computer Graphics	3	CSA	SNB
7	CS3461	Advanced Operating Systems	3	SY	KS
8	CS3471	Design and Analysis of Parallel Algorithms	3	TCS	KS
9	CS3481	Advanced Data Structures	3	TCS	KS
10	CS3491	Cloud Computing	3	CSA	KHB
11	CS3501	Model-driven Frameworks	3	CSA	KHB
12	CS3511	Cyber Laws and IPR	3	SP	SNB

SL.No.	Course Code	Sub Name	Credits	CAT	Course Designer
1	CS3701	Full Stack Development	3	CSA	SNB
2	CS3711	Virtual Reality and Augmented Reality	3	CSA	KHB



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH  
संगणक विज्ञान एवं अभियांत्रिकी विभाग  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

3	CS3721	Software Testing	3	CSA	SNB
4	CS3731	Wireless Technologies	3	CSA	CSP
5	CS3741	Social Network Analytics	3	DS	KHB
6	CS3751	Information Retrieval	3	DS	SNB
7	CS3761	Service Oriented Architecture	3	CSA	SNB
8	CS3771	Probabilistic Graphical Models	3	DS	KS
9	CS3781	Performance Modeling of Computer Systems	3	SY	KS
10	CS3791	Computational Number Theory	3	TCS	SNB
11	CS3801	Design Patterns	3	CSA	SNB
12	CS3811	Advanced Algorithms	3	TCS	KS
13	CS3821	Reinforcement Learning	3	DS	KS
14	CS3831	Deep Learning for Vision	3	DS	CHS
15	CS3841	Deep Learning for NLP	3	DS	KHB
16	CS3851	Generative AI	3	DS	SNB
17	CS3861	Explainable AI	3	DS	SNB
18	CS3871	High Performance Computing	3	SY	KS
19	CS3881	Game Theory	3	TCS	SNB
20	CS3891	Program Analysis and Verification	3	TCS	KS
21	CS3902	Deep Learning Lab	3	DS	SNB

VII – Semester, Department Electives								
SL. No.	Course Code	Course Title	L	T	P	Credits	Cat	Course Designer
1	CS4401	Privacy Preserving Data Processing	3	-	-	3	SP	SNB
2	CS4411	Human Computer Interaction	3	-	-	3	CSA	KHB
3	CS4421	Advanced Computational Statistics	3	-	-	3	DS	KS
4	CS4431	Real Time Systems	3	-	-	3	SY	KS
5	CS4441	Soft Computing	3	-	-	3	DS	CHS
6	CS4451	Applications of AI in Healthcare	2	-	-	2	DS	KS
7	CS4461	Green Computing	3	-	-	3	CSA	KS



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH  
संगणक विज्ञान एवं अभियांत्रिकी विभाग  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

8	CS4471	AI Ethics and Societal Challenges	3	-	-	3	DS	CSP
9	CS4481	Blockchains	3	-	-	3	SP	SNB
10	CS4491	Intrusion Detection Systems	3	-	-	3	SP	CSP
11	CS4501	Malware Detection and Mitigation	3	-	-	3	SP	CSP
12	CS4511	Advanced Computer Networks	3	-	-	3	SY	CSP
13	CS4521	Internet of Things	3	-	-	3	SY	CSP
14	CS4522	Internet of Things Lab	-	1	2	2	SY	CSP
15	CS4531	Computer Forensics and Cyber Security	3	-	-	3	SP	CSP
16	CS4541	Agile Methodologies	3	-	-	3	CSA	SNB
16	CS4551	Big Data Engineering	2	1	-	3	CSA	SNB
17	CS4462	Security Lab	-	1	2	2	SP	SNB

The following suggestions were offered by the BoS:

1. Data Mining and Data Warehousing courses must be refined to exclude Big Data and Hadoop tools to avoid redundancy. Data Mining Algorithms are to be included for improving the rigor of the course
2. Data Science should bring more rigor with the inclusion of matrix factorization, SVD and Link Analysis
3. Advanced Database Systems course content should be refined to include advanced topics on shared nothing architectures, data partitioning strategies, database security
4. Randomized Algorithms course content was to be divided into multiple paragraphs, rather than a single paragraph.
5. Recent research papers are to be included in the reading list in courses such as "Green Computing" and "Program Analysis and Verification".
6. Review components in the syllabi of Graph Algorithms and Advanced Computational Statistics are to be removed.
7. The courses are to be annotated with categorization of sub-areas of Computer Science into which each elective belongs.
8. GPU Programming has to be included in High Performance Computing Course.

The following table summarizes the revisions made in this regard.

S.NO	Annexure	Content Description	Comments
1	Annexure-F	II-B.Tech. II Sem DEC Courses	• It is same content as presented in the previous BOS.



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH  
संगणक विज्ञान एवं अभियांत्रिकी विभाग  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

			<ul style="list-style-type: none"><li>• There is no revision for course content for any of the courses, except for formatting and pre-requisite mapping</li></ul>
2	Annexure-G	III-B.Tech. I Sem DEC Courses	<ul style="list-style-type: none"><li>• The course contents of all the III-I DEC courses incorporating the BOS suggestions are presented in this Annexure.</li><li>• The dependencies of all DEC courses are presented pictorially in Annexure A</li></ul>
3	Annexure-H	III-B.Tech. II Sem DEC Courses	<ul style="list-style-type: none"><li>• The course contents of all the III-II DEC courses incorporating the BOS suggestions are presented in this Annexure.</li><li>• The dependencies of all DEC courses are presented pictorially in Annexure A</li></ul>
4	Annexure-I	IV-B.Tech. I Sem DEC Courses	<ul style="list-style-type: none"><li>• The course contents of all IV-II DEC courses incorporating the BOS suggestions are presented in this Annexure.</li><li>• The dependencies of all DEC courses are presented pictorially in Annexure A</li></ul>

**Resolution: The BOS members approved the revised curriculum contents of the courses which are presented in the Annexures F, G, H, I**



**Item No: 4**

**Sub:** To consider and approve minor/double major degree courses for the minor degree offerings with titles “Software Development”, “Artificial Intelligence and Machine Learning”, “Quantum Computing” and double major with title “Double Major in CSE”.

The overall course structure of minor degree with title “Software Development” is depicted below:

<b>Minor program Course Distribution (Minor in Software Development)</b>								
<b>SL. No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Offered Sem</b>	<b>Cat. Code</b>
1	CS2051	Data Base Management Systems	3	-	-	3	IV	PCC
2	CS2061	Software Development	2	1	-	3	IV	PCC
3		Slot – III				3	V	DEC
4		Slot – IV				3	VI	DEC
		<b>TOTAL</b>				<b>12</b>		

**DEC Courses for Minor in Software Development**

CS3551	Software Testing	3	-	-	3
CS2031	Design and Analysis of Algorithms	3	-	-	3
CS2032	Algorithms Lab	-	-	2	1
CS2052	Data Base Management System Lab	0	-	2	1
CS3571	Full Stack Development	3	-	-	3
CS3581	Model-driven Frameworks	3	-	-	3
CS3741	Service Oriented Architecture	3	-	-	3
CS2422	Mobile Application Development Lab	-	-	2	1
CS2432	Software Coding Standards Lab	-	-	2	1

The contents of these courses are part of the regular curriculum depicted in Annexure E.



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
**NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH**  
 संगणक विज्ञान एवं अभियांत्रिकी विभाग  
 DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

<b>Double Major program Course Distribution (Double Major in CSE)</b>									
S. No.	Course Code	Course Title	L	T	P	Credits	Offered Sem	Cat. Code	
1	CS2021	Operating Systems	3	-	-	3	III	PCC	
2	CS2031	Design and Analysis of Algorithms	3	-	-	3	III	PCC	
3	CS2051	Data Base Management Systems	3	-	-	3	IV	PCC	
4	CS2061	Software Development	2	1	-	3	IV	PCC	
5		Slot – III				6	V	DEC	
6		Slot – IV				6	VI	DEC	
		<b>TOTAL</b>				<b>24</b>			

Slot-III and Slot-IV credits can be taken from any of the department electives not covered under regular B.Tech. program credits.

<b>Minor program Course Distribution (Minor in Artificial Intelligence and Machine Learning)</b>									
S. No.	Course Code	Course Title	L	T	P	Credits	Offered Sem	Cat. Code	
1	CS2031	Artificial Intelligence	3	-	-	3	IV	PCC	
2	CS2061	Machine Learning	3	-	-	3	V	PCC	
3		Slot – III				3	V	DEC	
4		Slot – IV				3	VI	DEC	
		<b>TOTAL</b>				<b>12</b>			

**DEC options for minor degree**

CS3411	<b>Data Science</b>	3	-	-	3
CS3501	Probabilistic Graphical Models	3	-	-	3
CS3511	Semantic Web	3	-	-	3
CS3721	Social Network Analytics	3	-	-	3
CS3731	Information Retrieval	3	-	-	3
CS3801	Reinforcement Learning	3	-	-	3
CS3811	Deep Learning for Vision	3	-	-	3
CS3821	Deep Learning for NLP	3	-	-	3
CS3831	Generative AI	3	-	-	3
CS3841	Explainable AI	3	-	-	3



**राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश**  
**NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH**  
**संगणक विज्ञान एवं अभियांत्रिकी विभाग**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

CS4441	Soft Computing	3	-	-	3
CS4451	Applications of AI in Healthcare	-	1	2	2
CS4462	Deep Learning lab	2	-	-	2
CS4471	AI Ethics and Societal Challenges	2	-	-	2
CS2402	Data Processing and Visualization lab	-	-	2	1
CS2412	Modelling and Optimization lab	-	-	2	1

<b>Minor program Course Distribution (Minor in Quantum Computing)</b>								
SL. No.	Course Code	Course Title	L	T	P	Credits	Offered Sem	Cat. Code
1	CS2131	Foundations of Quantum Computing	2	-	-	2	IV	PCC
2	PH2XX1	Foundations of Quantum Physics	2	-	-	2	IV	PCC
3	EC2711	Foundations of Quantum Sensing and Communication	2	-	-	2	IV	PCC
4	CS3302	Basic Quantum Computing Programming Lab	0	2	2	3	V	PCC
5		Slot-V	3	-	-	3	V	DEC
6		Slot-VI	3	-	-	3	VI	DEC
		TOTAL				18		

**DEC options for minor degree**

1	*CS3311	Introduction to Quantum Computation	3	0	0	3	V	DEC
2	*EC3711	Introduction to Quantum Communication	3	0	0	3	V	DEC
3	*PH3XX1	Introduction to Quantum Materials	3	0	0	3	V	DEC
4	PH3XX1	Solid State Physics for Quantum Technologies	3	0	0	3	VI	DEC

- One of these courses is to be offered in Slot-V.

**Resolution: The BOS members approved the curriculum contents of the minor and double major courses which are presented in the Annexure A and K.**

**Members Present:**



राष्ट्रीय प्रौद्योगिकी संस्थान- आंध्रप्रदेश  
NATIONAL INSTITUTE OF TECHNOLOGY, ANDHRA PRADESH  
संगणक विज्ञान एवं अभियांत्रिकी विभाग  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

---

S. Karthick  
25/06/2025

Dr. Karthick S

K. Hima Bindu  
25/06/25

Dr. K. Hima Bindu

S. Srilatha  
25/06/2025

Dr. Srilatha Chebrolu

Choudhary Shyam Prakash  
26/06/25

Dr. Choudhary Shyam Prakash

R.B.V. Subramanyam

Prof. R.B.V. Subramanyam

Mary Saira Bhanu

Prof. Mary Saira Bhanu

Sridhar Chimalakonda

Dr. Sridhar Chimalakonda

K. Venkateswara Rao

Dr. Kollaparthi Venkateswara Rao

CH N A B Sankar  
28/6/25

Dr. CH N A B Sankar

Mr. Abhijat Ande

Dr. N. Jayaram

Dr. R. Sunil Kumar

S. Nagesh Bhattu

Dr. S. Nagesh Bhattu  
Chairman

**NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

About the Department of CSE

The Department of Computer Science and Engineering at NIT Andhra Pradesh, offers a B.Tech. Undergraduate programme in Computer Science and Engineering, M. Tech in Computer Science and Data Analytics, M.S. (by research) and Ph.D. programmes. The department was incepted in the year 2015, the current sanctioned intake is 90 students for the B.Tech. (CSE) programme.

Vision of the Department of CSE

To strive for excellence in academics, research and technological service to the society, with an intent to nurture the stakeholders and produce Computer Scientists, Technologists and Engineers who are globally competent and nationally relevant.

Mission of the Department of CSE

- M 1. To adopt a teaching-learning process that imparts technical skills and state-of the-art knowledge with a well-blended and balanced mix of theory and practice.
- M 2. To create functional centres of excellence that promote research and consultancy in the thrust sub-domains of theoretical computer science, systems and technology.
- M 3. To collaborate with industry and higher learning institutes of national/international repute and solve socially relevant problems

Programme Educational Objectives (PEOs) for the B.Tech. (CSE) Programme:

Within few years after the end of the B.Tech. in Computer Science and Engineering programme, graduates will be able to:

<b>PEO1</b>	Effectively and efficiently function as Software Engineers and Technologists in organizations that deliver Computing or Information Technology based innovations, products and solutions.
<b>PEO2</b>	Enhance academic and research credentials by pursuing specialized postgraduate and / or doctoral education in institutes of higher learning in the niche sub-domains of theoretical Computer Science, Systems, Technology and Applications in the sub-disciplines including but not limited to AI, Data Analytics, Machine Learning, Security and Distributed Computing.
<b>PEO3</b>	Engage in continuous learning and skill enhancement to adapt and contribute to the global and national needs in terms of computing or IT based innovative solutions with a sincere practice of professional ethics and technical quality to meet or surpass the requirements of the stakeholders.

Programme Articulation Matrix (PEO vs. Mission) for the B.Tech. (CSE) Programme:

<b>PEO\Mission</b>	<b>M1</b>	<b>M2</b>	<b>M3</b>
<b>PEO1</b>	S	M	S
<b>PEO2</b>	S	S	M
<b>PEO3</b>	M	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

## Programme Outcomes (POs) for the B.Tech. (CSE) Programme

At the end of the program, the student will be able to:

<b>PO1</b>	<b>Engineering knowledge:</b> Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems
<b>PO2</b>	<b>Problem analysis:</b> Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
<b>PO3</b>	<b>Design/Development of solutions:</b> Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
<b>PO4</b>	<b>Conduct investigations of complex problems:</b> Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
<b>PO5</b>	<b>Modern tool usage:</b> Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
<b>PO6</b>	<b>The engineer and society:</b> Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
<b>PO7</b>	<b>Environment and sustainability:</b> Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
<b>PO8</b>	<b>Ethics:</b> Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
<b>PO9</b>	<b>Individual and team work:</b> Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
<b>PO10</b>	<b>Communication:</b> Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
<b>PO11</b>	<b>Project management and Finance:</b> Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

<b>PO12</b>	<b>Life-long learning:</b> Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
-------------	---

### Programme Specific Outcomes (PSO) for the B.Tech. (CSE) Programme

At the end of the B.Tech. in Computer Science and Engineering programme, graduates will be able to:

<b>PSO1</b>	Design, analyze, implement, verify and validate efficient solutions to complex engineering problems related to the ideation, development, testing and maintenance of computing systems.
<b>PSO2</b>	Construct or leverage contemporary tools, techniques, and frameworks in developing or refactoring a computing system or its component.
<b>PSO3</b>	Apply research-based methods to construct, implement, verify and validate analytical and simulation models for addressing generative, predictive, diagnostic and prescriptive tasks related to Natural Language Processing, Computer Vision, Automation and Security.

### Credit Distribution

Category	Category Description	Credits	Percentage
BSC	Basic Science Courses (BSC)	15	10%
ESC	Engineering Science Courses (ESC)	15	10%
Professional Major Courses (PMC)	Professional Major Core Courses (PCC)	60	60%
	Professional Major Elective Courses (DEC)	18	
	Professional Major Work (PRC)	6	
	Semester-Long Internship (SLI)	6	
Open/Free Electives	Open Elective Courses (OEC) and DAC approved Free Electives (NPTEL, MOOCs, etc.)	15	10%
HSC Courses	Liberal Arts/Creative Arts Courses (LCA)	6	10%
	Sports Courses (Any two sports of 1 Credit each)	2	
	NCC/Social Service	1	
	Yoga	1	
	English Communication	2	
	Personality Development/Life Skills	1	
	Introduction to Entrepreneurship	1	
	Introduction to Design Thinking	1	
	<b>Total Credits</b>	<b>150</b>	<b>100%</b>

## B.Tech. Computer Science and Engineering Course Structure

### I Year B.Tech. Course Structure

<b>I – Year: I – Semester</b>							
<b>S.No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Cat. Code</b>
1	MA1011	Differential and Integral Calculus	3	-	-	3	BSC
2	PH1011	Engineering Physics	2	-	-	2	BSC
3	PH1012	Engineering Physics Lab	-	-	2	1	BSC
4	CS1011	Introduction to Computational Problem Solving	3	-	-	3	ESC
5	CS1012	Introduction to Computational Problem-Solving Lab	-	1	2	2	ESC
6	EC1531	Digital Logic Design	3	-	-	3	ESC
7	CS1021	Computer Organization and Architecture	3	-	-	3	PCC
8	HS1022	Physical Education	-	-	2	1	HSC
		Total				18	
<b>I – Year: II – Semester</b>							
<b>S.No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Cat. Code</b>
1	MA1021	Matrices and Differential Equations	3	-	-	3	BSC
2	MA1051	Discrete Mathematics	3	-	-	3	BSC
3	CS1051	Data Structures and Algorithms	3	-	-	3	PCC
4	CS1052	Data Structures and Algorithms Lab	-	1	2	2	PCC
5	CS1061	Object Oriented Design and Programming	3	-	-	3	PCC
6	CS1062	Object Oriented Design and Programming Lab	-	-	2	1	PCC
7	HS1011	English for Engineers – I	2	-	-	2	HSC
8	HS1032	Health Education	-	-	2	1	HSC
		Total				18	

II Year B.Tech. CSE Course Structure

<b>II – Year: I – Semester</b>							
<b>S.No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Cat. Code</b>
1	MA2031	Probability, Statistics and Stochastic Processes	3	-	-	3	BSC
2	EC1541	Microprocessors	2	-	-	2	ESC
3	EC1542	Microprocessors Lab	-	1	2	2	ESC
4	CS2011	Discrete Event Simulation	1	-	-	1	ESC
5	EC2511	Modelling and Optimization Techniques	2	-	-	2	ESC
6	CS2021	Operating Systems	3	-	-	3	PCC
7	CS2022	Operating Systems Lab	-	1	2	2	PCC
8	CS2031	Design and Analysis of Algorithms	3	-	-	3	PCC
9	CS2032	Algorithms Lab	-	-	2	1	PCC
10	CS2041	Theory of Computation	3	-	-	3	PCC
11	HS1052	Social Service	-	-	2	1	HSC
		<b>Total</b>				<b>23</b>	
<b>II – Year: II – Semester</b>							
<b>S.No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Cat. Code</b>
1	CS2051	Data Base Management Systems	3	-	-	3	PCC
2	CS2052	Data Base Management Systems Lab	-	-	2	1	PCC
3	CS2061	Software Development	2	1	-	3	PCC
4	CS2062	Software Development Lab	-	-	2	1	PCC
5	CS2071	Compiler Design	3	-	-	3	PCC
6	CS2072	Compiler Design Lab	-	-	2	1	PCC
7	CS2081	Artificial Intelligence	3	-	-	3	PCC
8		Department Elective				1	DEC
9		Open Elective /DAC approved Free Electives (NPTEL, MOOCs, etc.)	3	-	-	3	OEC
10	HS2011	Personality Development	1	-	-	1	HSC
11	PE2012	Yoga	-	-	2	1	HSC
		<b>Total</b>				<b>21</b>	

**II – Year: II – Semester, Department Electives\***

<b>S.No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>
1	CS2402	Data Processing and Visualization Lab	-	-	2	1
2	CS2412	Modelling and Optimization Lab	-	-	2	1
3	CS2422	Mobile Application Development Lab	-	-	2	1
4	CS2432	Software Coding Standards Lab	-	-	2	1

\*Additional titles can be offered as per the availability of faculty after BOS approval of the curriculum for additional titles

### III Year B.Tech. CSE Course Structure

<b>III – Year: I – Semester</b>							
<b>S.No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Cat. Code</b>
1	CS3011	Machine Learning	3	-	-	3	PCC
2	CS3012	Machine Learning Lab	-	-	2	1	PCC
3	CS3021	Software Engineering	2	-	-	2	PCC
4	CS3022	Software Engineering Lab	-	-	2	1	PCC
5	CS3031	Computer Networks	2	1	-	3	PCC
6	CS3032	Computer Networks Lab	-	-	2	1	PCC
7		Department Electives				6	DEC
8		Open Elective /DAC approved Free Electives (NPTEL, MOOCs, etc.)	3	-	-	3	OEC
9	SM3021	Introduction to Design Thinking				1	HSC
		<b>Total</b>				<b>21</b>	

<b>V – Semester, Department Electives*</b>							
<b>S.No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	
1	CS3401	Data Warehousing and Data Mining	3	-	-	3	
2	CS3411	Data Science	3	-	-	3	
3	CS3421	Advanced Database Systems	3	-	-	3	
4	CS3431	Graph Algorithms	3	-	-	3	
5	CS3441	Randomized Algorithms	3	-	-	3	
6	CS3451	Computer Graphics	3	-	-	3	
7	CS3461	Advanced Operating Systems	3	-	-	3	
8	CS3471	Design and Analysis of Parallel Algorithms	3	-	-	3	
9	CS3481	Advanced Data Structures	3	-	-	3	
10	CS3491	Cloud Computing	3	-	-	3	
11	CS3501	Model-Driven Frameworks	3	-	-	3	
12	CS3511	Cyber Laws and IPR	3	-	-	3	

\*Additional titles can be offered as per the availability of faculty after BOS approval of the curriculum for additional titles

### III – Year: II – Semester

S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1	CS3051	Distributed Computing	2	1	-	3	PCC
2	CS3052	Distributed Computing Lab	-	-	2	1	PCC
3	CS3061	Cryptography and Network Security	3	-	-	3	PCC
4	CS3071	Theory and Design of Programming Languages	2	1	-	3	PCC
5		Department Electives				6	DEC
6		Open Elective /DAC approved Free Electives (NPTEL, MOOCs, etc.)	3	-	-	3	OEC
7	HS3011	English for Engineers – II(LCA-I)	2	1	-	3	HSC
8	SM3011	Introduction to Entrepreneurship				1	HSC
		Total				23	

### VI – Semester, Department Electives\*

S.No.	Course Code	Course Title	L	T	P	Credits
1	CS3701	Full Stack Development	3	-	-	3
2	CS3711	Virtual Reality and Augmented Reality	3	-	-	3
3	CS3721	Software Testing	3	-	-	3
4	CS3731	Wireless Technologies	3	-	-	3
5	CS3741	Social Network Analytics	3	-	-	3
6	CS3751	Information Retrieval	3	-	-	3
7	CS3761	Service Oriented Architecture	3	-	-	3
8	CS3771	Probabilistic Graphical Models	3	-	-	3
9	CS3781	Performance Modeling of Computer Systems	3	-	-	3
10	CS3791	Computational Number Theory	3	-	-	3
11	CS3801	Design Patterns	3	-	-	3
12	CS3811	Advanced Algorithms	3	-	-	3
13	CS3821	Reinforcement Learning	3	-	-	3
14	CS3831	Deep Learning for Vision	3	-	-	3
15	CS3841	Deep Learning for NLP	3	-	-	3
16	CS3851	Generative AI	3	-	-	3
17	CS3861	Explainable AI	3	-	-	3
18	CS3871	High Performance Computing	3	-	-	3
19	CS3882	Game Theory	3	-	-	3
20	CS3891	Program Analysis and Verification	3	-	-	3
21	CS3902	Deep Learning Lab	0	1	2	2

\*Additional titles can be offered as per the availability of faculty after BOS approval of the curriculum for additional titles

IV Year B.Tech. CSE Course Structure

IV – Year: I – Semester							
S.No.	Course Code	Course Title	L	T	P	Credits	Cat. Code
1		Open Elective /DAC approved Free Electives (NPTEL, MOOCs, etc.)	6	-	-	6	OEC
2	CS4014	Professional Major Work	-	-	12	6	PRC
3		Department Electives				5	DEC
4		Liberal Arts/Creative Arts Courses – II				3	HSC
		Total				20	

VII – Semester, Department Electives*							
S.No.	Course Code	Course Title	L	T	P	Credits	
1	CS4401	Privacy Preserving Data Processing	3	-	-	3	
2	CS4411	Human Computer Interaction	3	-	-	3	
3	CS4421	Advanced Computational Statistics	3	-	-	3	
4	CS4431	Real Time Systems	3	-	-	3	
5	CS4441	Soft Computing	3	-	-	3	
6	CS4451	Applications of AI in Healthcare	2	-	-	2	
7	CS4461	Green Computing	3	-	-	3	
8	CS4471	AI Ethics and Societal Challenges	2	-	-	2	
9	CS4481	Blockchains	3	-	-	3	
10	CS4491	Intrusion Detection Systems	3	-	-	3	
11	CS4501	Malware Detection and Mitigation	3	-	-	3	
12	CS4511	Advanced Computer Networks	3	-	-	3	
13	CS4521	Internet of Things	3	-	-	3	
14	CS4522	Internet of Things Lab	2	-	-	2	
16	CS4531	Computer Forensics & Cyber Security	3	-	-	3	
17	CS4541	Agile Methodologies	3	-	-	3	
18	CS4551	Big Data Engineering	2	1	-	3	
19	CS4562	Security Lab	-	1	2	2	

\*Additional titles can be offered as per the availability of faculty after BOS approval of the curriculum for additional titles

**IV – Year: II – Semester**

<b>S.No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Cat. Code</b>
1		Semester-Long Internship (SLI) /Additional Project at the institute/Additional department elective courses for 6 credits	-	-	12	6	SLI
		Total				6	

## Minor/Double Major programs

- To get Minor degree, a student must earn 12 Credits (6 Core credits + 6 Elective Credits) prescribed for the programme.
- To get Double Major degree, a student must earn 24 Credits (12 Core credits + 12 Elective Credits) prescribed for the programme.
- The Minor & Double Major choices start from the beginning of 3<sup>rd</sup> Semester.
- The students of Minor & Double Major courses will sit with regular students in the class.
- There will separate timetable slots for Minor & Double Major courses. Courses other than Minor & Double Major courses will not be offered during these slots.
- A student is permitted to do one Minor and one Double Major at max.
- The department will choose a CGPA cut off (based on II Sem CGPA) such that the total of Minor and Double Major students in any branch do not cross maximum of 30 seats.

<b>Minor program Course Distribution (Minor in Software Development)</b>								
<b>S. No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Offered Sem</b>	<b>Cat. Code</b>
1	CS2051	Data Base Management Systems	3	-	-	3	IV	PCC
2	CS2061	Software Development	2	1	-	3	IV	PCC
3		Slot – III				3	V	DEC
4		Slot – IV				3	VI	DEC
		<b>TOTAL</b>				<b>12</b>		

### **DEC Courses for Minor in Software Development**

CS3551	<b>Software Testing</b>	3	-	-	3
CS2031	Design and Analysis of Algorithms	3	-	-	3
CS2032	Algorithms lab	-	-	2	1
CS2052	Data Base Management System Lab	0	-	2	1
CS3571	Full Stack Development	3	-	-	3
CS3581	Model-driven Frameworks	3	-	-	3
CS3761	Service Oriented Architecture	3	-	-	3
CS2422	Mobile Application Development lab	-	-	2	1
CS2432	Software Coding Standards lab	-	-	2	1

<b>Double Major program Course Distribution (Double Major in CSE)</b>								
<b>S. No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Offered Sem</b>	<b>Cat. Code</b>
1	CS2021	Operating Systems	3	-	-	3	III	PCC
2	CS2031	Design and Analysis of Algorithms	3	-	-	3	III	PCC
3	CS2051	Data Base Management Systems	3	-	-	3	IV	PCC
4	CS2061	Software Development	2	1	-	3	IV	PCC
5		Slot – III				6	V	DEC
6		Slot – IV				6	VI	DEC
		TOTAL				24		

Slot-III and Slot-IV credits can be taken from any of the department electives not covered under regular B.Tech. program credits.

<b>Minor program Course Distribution (Minor in Artificial Intelligence and Machine Learning)</b>								
<b>S. No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Offered Sem</b>	<b>Cat. Code</b>
1	CS2031	Artificial Intelligence	3	-	-	3	IV	PCC
2	CS2061	Machine Learning	3	-	-	3	V	PCC
3		Slot – III				3	V	DEC
4		Slot – IV				3	VI	DEC
		TOTAL				12		

**DEC options for minor degree**

CS3411	Data Science	3	-	-	3
CS3501	Probabilistic Graphical Models	3	-	-	3
CS3511	Semantic Web	3	-	-	3
CS3721	Social Network Analytics	3	-	-	3
CS3731	Information Retrieval	3	-	-	3
CS3801	Reinforcement Learning	3	-	-	3
CS3811	Deep Learning for Vision	3	-	-	3
CS3821	Deep Learning for NLP	3	-	-	3
CS3831	Generative AI	3	-	-	3

CS3841	Explainable AI	3	-	-	3
CS4441	Soft Computing	3	-	-	3
CS4451	Applications of AI in Healthcare	-	1	2	2
CS4462	Deep Learning Lab	2	-	-	2
CS4471	AI Ethics and Societal Challenges	2	-	-	2
CS2402	Data Processing and Visualization Lab	-	-	2	1
CS2412	Modelling and Optimization Lab	-	-	2	1

<b>Minor program Course Distribution (Minor in Quantum Computing)</b>								
<b>S. No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Offered Sem</b>	<b>Cat. Code</b>
1	CS2131	Foundations of Quantum Computing	2	-	-	2	IV	PCC
		Foundations of Quantum Physics	2	-	-	2	IV	PCC
		Foundations of Quantum Sensing and Communication	2	-	-	2	IV	PCC
2	CS2141	Basic Quantum Computing Programming Lab	2	0	2	3	IV	PCC
3	CS3081	Introduction to Quantum Computation	3	0	0	3	V	DEC
		Introduction to Quantum Communication	3	0	0	3	V	DEC
		Introduction to Quantum Materials	3	0	0	3	V	DEC
4		Solid State Physics for Quantum Technologies	3	0	0	3	VI	DEC
		<b>TOTAL</b>				<b>18</b>		

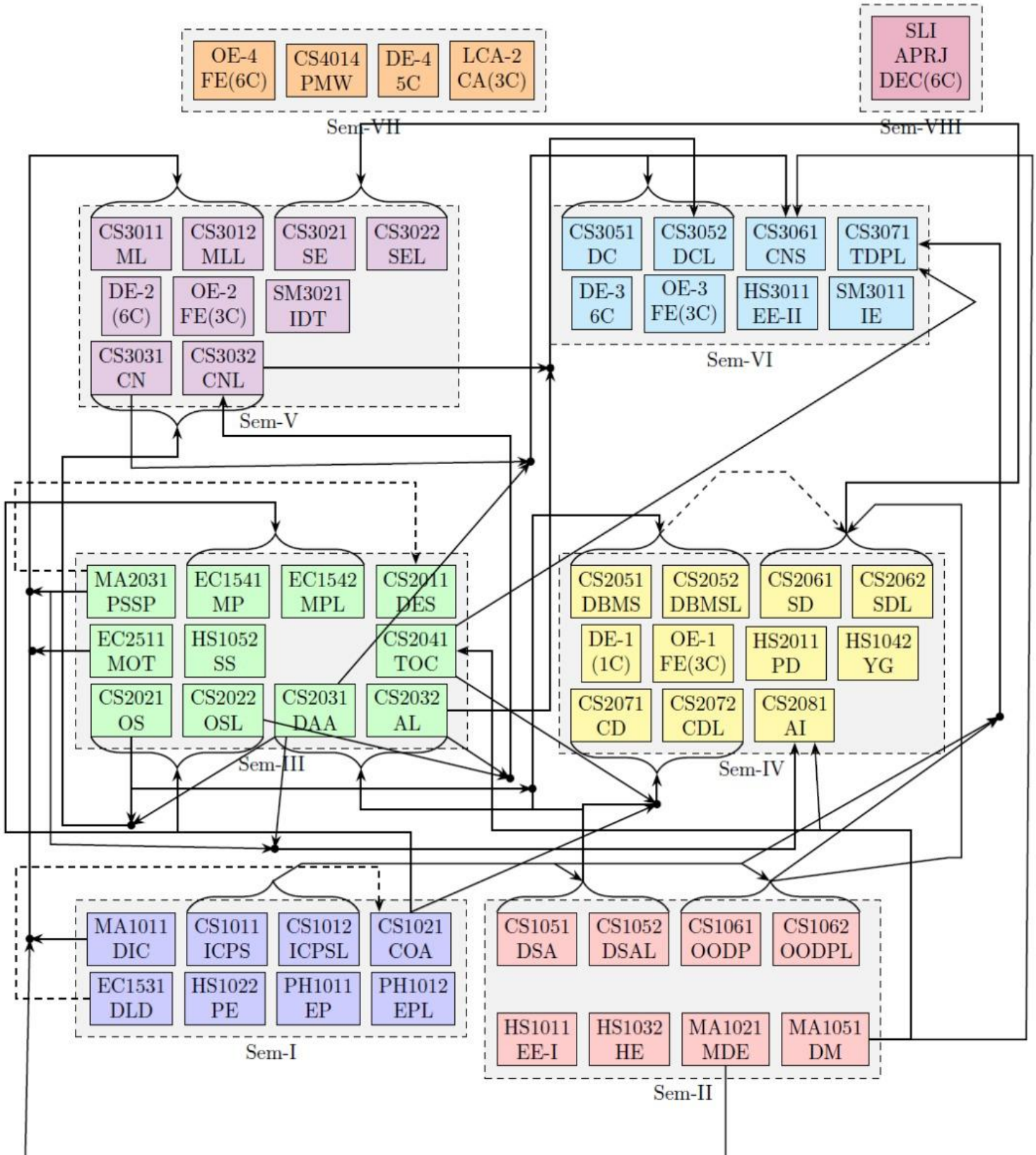
Engineering Science Courses offered by DCSE

<b>Engineering Science Courses</b>								
<b>S. No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Cat. Code</b>	
1	CS1011	Introduction to Computational Problem Solving	3	-	-	3	ESC	
2	CS1012	Introduction to Computational Problem-Solving Lab	-	1	2	2	ESC	
3	CS1031	Problem Solving through Computer Programming	3	-	-	3	ESC	
4	CS1032	Problem Solving through Computer Programming Lab	-	1	2	2	ESC	
5	CS1041	Introduction to Computer Programming	3	-	-	3	ESC	
6	CS1042	Introduction to Computer Programming Lab	-	1	2	2	ESC	
7	CS2101	Data Structures and Applications	3	-	-	3	ESC	
8	CS2102	Data Structures and Applications Lab	-	-	2	1	ESC	
9	CS2011	Discrete Event Simulation	1	-	-	1	ESC	
10	CS2121	Basics of Algorithm Design	1	-	-	1	ESC	

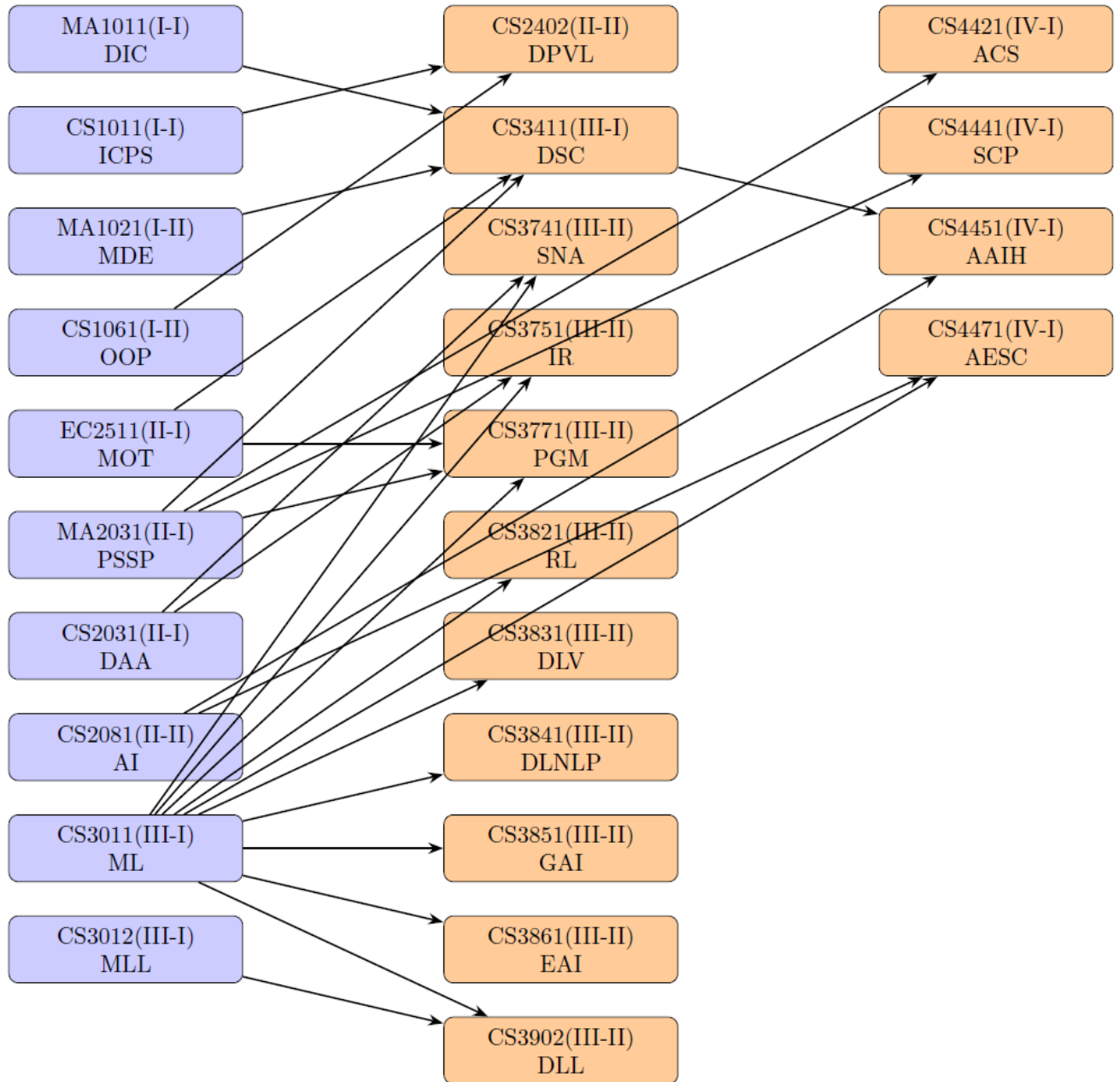
Open Elective Courses offered by DCSE

<b>OPEN ELECTIVE COURSES</b>								
<b>S. No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Offered Sem</b>	<b>Cat. Code</b>
1	CS2301	Object Oriented Programming	2	1	-	3	IV	OEC
2	CS3301	Database Applications	2	1	-	3	V	OEC
3	CS3311	Web Programming	2	1	-	3	VI	OEC
4	CS4301	AI for Engineers	2	1	-	3	VII	OEC
5	CS4311	Numerical Computing with Python	3	-	-	3	VII	OEC

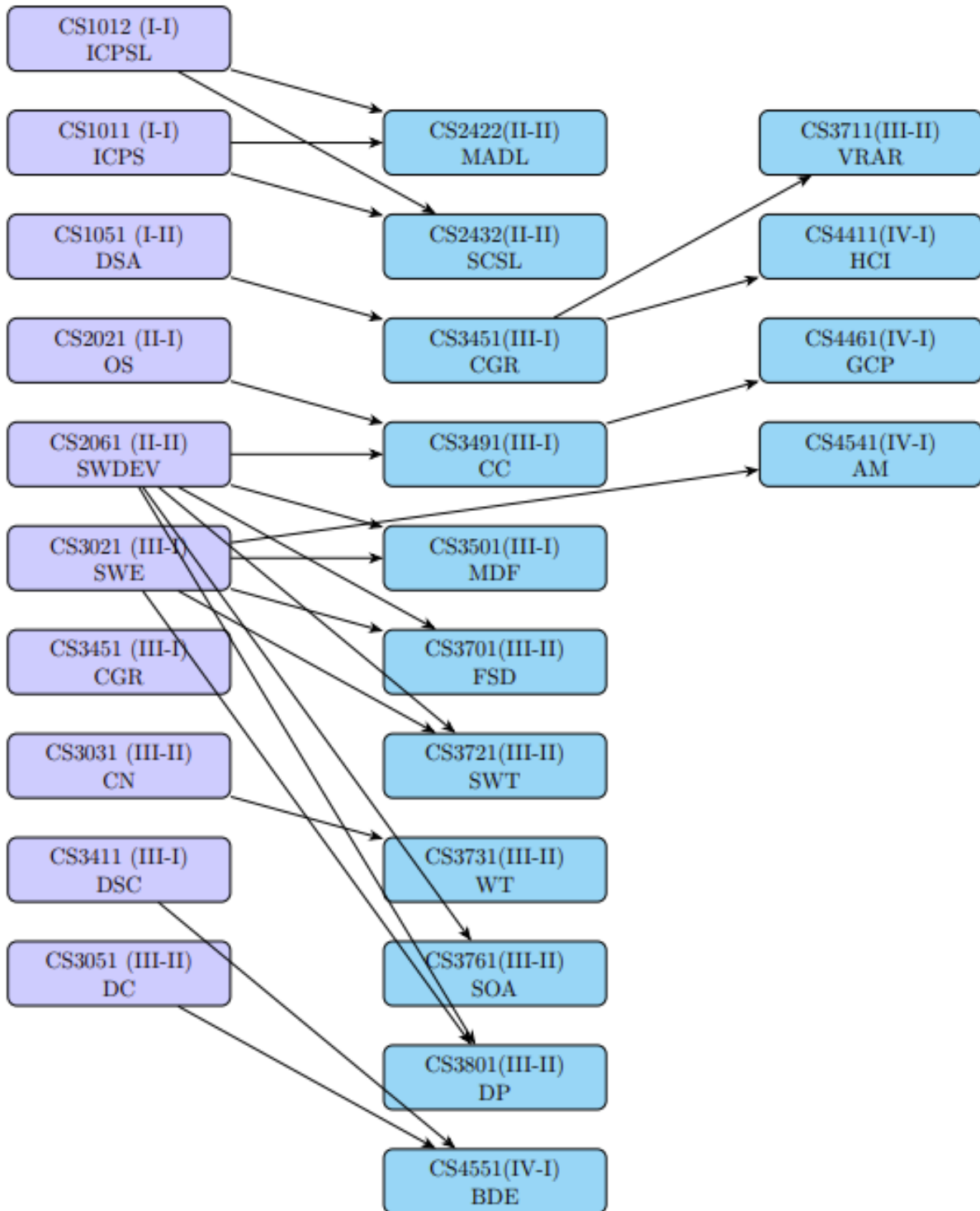
# 1 Dependencies of Core Courses



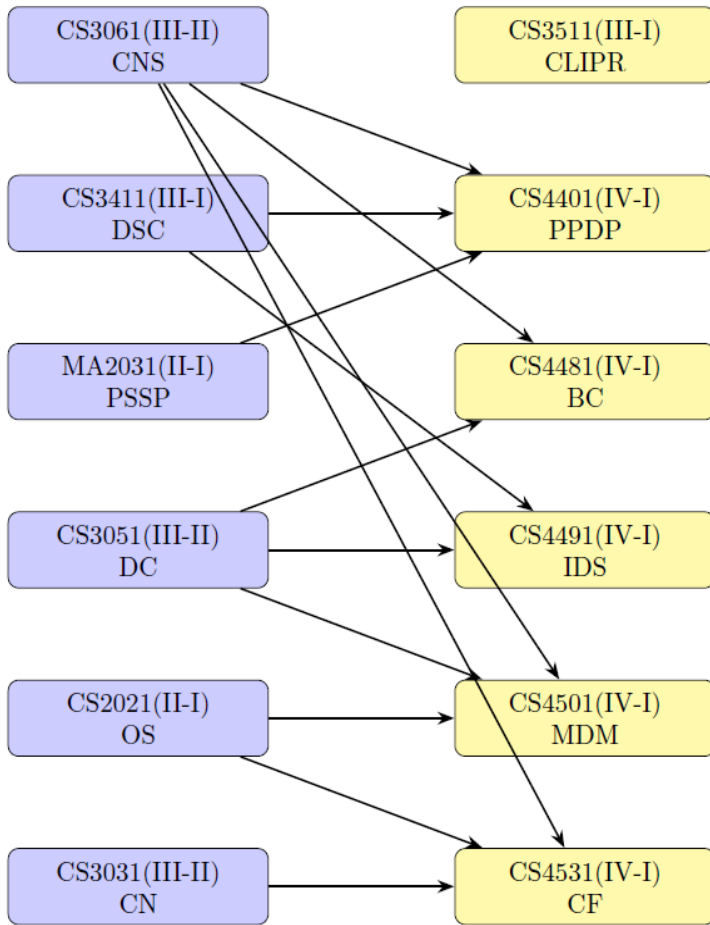
## 2 Dependencies of Department Electives under the sub-category Data Science(DS)



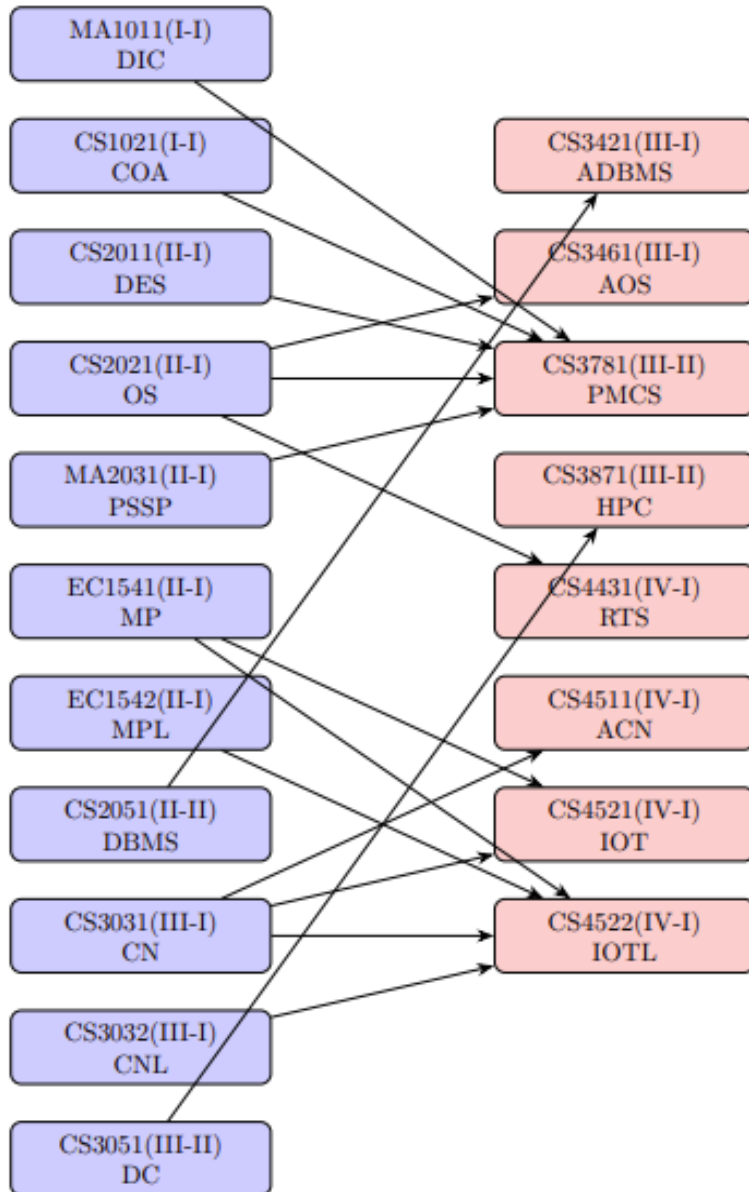
### 3 Dependencies of Department Electives in the sub-category Computer Science Applications(CSA)



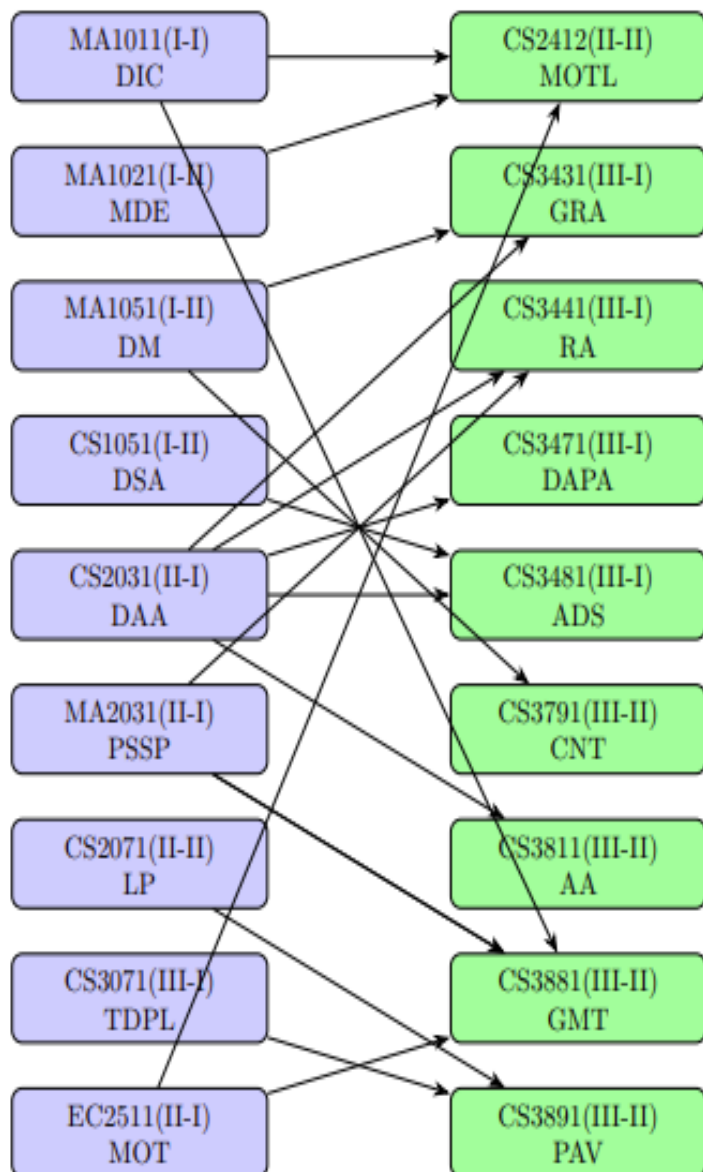
#### 4 Dependencies of Department Electives for courses under the sub-category Security and Privacy (SP)



## 5 Dependencies of Department Electives under the sub-category Systems(SY)



## 6 Dependencies of Electives under the sub-category Theoretical Computer Science (TCS)



Course Code	Course Code Acronym	Course Name	Sub Category for DEC Courses
<b>CS1011</b>	ICPS	Introduction to Computational Problem Solving	
<b>CS1012</b>	ICPSL	Introduction to Computational Problem-Solving Lab	
<b>CS1021</b>	COA	Computer Organization and Architecture	
<b>EC1531</b>	DLD	Digital Logic Design	
<b>HS1022</b>	PHE	Physical Education	
<b>MA1011</b>	DIC	Differential and Integral Calculus	
<b>PH1011</b>	EPH	Engineering Physics	
<b>PH1012</b>	EPHL	Engineering Physics Lab	
<b>CS1051</b>	DSA	Data Structures and Algorithms	
<b>CS1052</b>	DSAL	Data Structures and Algorithms Lab	
<b>CS1061</b>	OOP	Object Oriented Design and Programming	
<b>CS1062</b>	OOPL	Object Oriented Design and Programming Lab	
<b>HS1011</b>	EE1	English for Engineers - I	
<b>HS1032</b>	HED	Health Education	
<b>MA1021</b>	MDE	Matrices and Differential Equations	
<b>MA1051</b>	DM	Discrete Mathematics	
<b>CS2011</b>	DES	Discrete Event Simulation	
<b>CS2021</b>	OS	Operating Systems	
<b>CS2022</b>	OSL	Operating Systems Lab	
<b>CS2031</b>	DAA	Design and Analysis of Algorithms	
<b>CS2032</b>	AGL	Algorithms Lab	
<b>CS2041</b>	TOC	Theory of Computation	
<b>EC1541</b>	MP	Microprocessors	
<b>EC1542</b>	MPL	Microprocessors Lab	
<b>EC2511</b>	MOT	Modelling and Optimization Techniques	
<b>HS1052</b>	SS	Social Service	
<b>MA2031</b>	PSSP	Probability, Statistics and Stochastic Processes	
<b>CS2051</b>	DBMS	Data Base Management Systems	
<b>CS2052</b>	DBMSL	Data Base Management Systems Lab	
<b>CS2061</b>	SWDEV	Software Development	

<b>CS2062</b>	SWDEVL	Software Development Lab	
<b>CS2071</b>	CD	Compiler Design	
<b>CS2072</b>	CDL	Compiler Design Lab	
<b>CS2081</b>	AI	Artificial Intelligence	
<b>CS2402</b>	DPVL	Data Processing and Visualization Lab	DS
<b>CS2412</b>	MOTL	Modelling and Optimization Lab	TCS
<b>CS2422</b>	MADL	Mobile Application Development Lab	CSA
<b>CS2432</b>	SCSL	Software Coding Standards Lab	CSA
<b>HS1042</b>	YG	Yoga	
<b>HS2011</b>	PDEV	Personality Development	
<b>CS3011</b>	ML	Machine Learning	
<b>CS3012</b>	MLL	Machine Learning Lab	
<b>CS3021</b>	SWE	Software Engineering	
<b>CS3022</b>	SWEL	Software Engineering Lab	
<b>CS3071</b>	TDPL	Theory and Design of Programming Languages	
<b>CS3401</b>	DWDM	Data Warehousing and Data Mining	
<b>CS3411</b>	DSC	Data Science	DS
<b>CS3421</b>	ADBMS	Advanced Database Systems	SY
<b>CS3431</b>	GRA	Graph Algorithms	TCS
<b>CS3441</b>	RA	Randomized Algorithms	TCS
<b>CS3451</b>	CGR	Computer Graphics	CSA
<b>CS3461</b>	AOS	Advanced Operating Systems	SY
<b>CS3471</b>	DAPA	Design and Analysis of Parallel Algorithms	TCS
<b>CS3481</b>	ADS	Advanced Data Structures	TCS
<b>CS3491</b>	CC	Cloud Computing	CSA
<b>CS3501</b>	MDF	Model-driven Frameworks	CSA
<b>CS3511</b>	CLIPR	Cyber Laws and IPR	SP
<b>CS3031</b>	CN	Computer Networks	
<b>CS3032</b>	CNL	Computer Networks Lab	
<b>CS3051</b>	DC	Distributed Computing	
<b>CS3052</b>	DCL	Distributed Computing Lab	
<b>CS3061</b>	CNS	Cryptography and Network Security	
<b>CS3701</b>	FSD	Full Stack Development	CSA
<b>CS3711</b>	VRAR	Virtual Reality and Augmented Reality	CSA
<b>CS3721</b>	SWT	Software Testing	CSA
<b>CS3731</b>	WT	Wireless Technologies	CSA
<b>CS3741</b>	SNA	Social Network Analytics	DS
<b>CS3751</b>	IR	Information Retrieval	DS
<b>CS3761</b>	SOA	Service Oriented Architecture	CSA
<b>CS3771</b>	PGM	Probabilistic Graphical Models	DS
<b>CS3781</b>	PMCS	Performance Modeling of Computer Systems	SY
<b>CS3791</b>	CNT	Computational Number Theory	TCS
<b>CS3801</b>	DP	Design Patterns	CSA

<b>CS3811</b>	AA	Advanced Algorithms	TCS
<b>CS3821</b>	RL	Reinforcement Learning	DS
<b>CS3831</b>	DLV	Deep Learning for Vision	DS
<b>CS3841</b>	DLNLP	Deep Learning for NLP	DS
<b>CS3851</b>	GAI	Generative AI	DS
<b>CS3861</b>	EAI	Explainable AI	DS
<b>CS3871</b>	HPC	High Performance Computing	SY
<b>CS3881</b>	GMT	Game Theory	TCS
<b>CS3891</b>	PAV	Program Analysis and Verification	TCS
<b>CS3902</b>	DLL	Deep Learning Lab	DS
<b>CS4551</b>	BDE	Big Data Engineering	CSA
<b>CS4401</b>	PPDP	Privacy Preserving Data Processing	SP
<b>CS4411</b>	HCI	Human Computer Interaction	CSA
<b>CS4421</b>	ACS	Advanced Computational Statistics	DS
<b>CS4431</b>	RTS	Real Time Systems	SY
<b>CS4441</b>	SCP	Soft Computing	DS
<b>CS4451</b>	AAIH	Applications of AI in Healthcare	DS
<b>CS4461</b>	GCP	Green Computing	CSA
<b>CS4471</b>	AESC	AI Ethics and Societal Challenges	DS
<b>CS4481</b>	BC	Blockchains	SP
<b>CS4491</b>	IDS	Intrusion Detection Systems	SP
<b>CS4501</b>	MDM	Malware Detection and Mitigation	SP
<b>CS4511</b>	ACN	Advanced Computer Networks	SY
<b>CS4521</b>	IOT	Internet of Things	SY
<b>CS4522</b>	IOTL	Internet of Things Lab	SY
<b>CS4531</b>	CF	Computer Forensics & Cyber Security	SP
<b>CS4541</b>	AM	Agile Methodologies	CSA

## Annexure-B

### Engineering Science Courses - Syllabi

<b>Engineering Science Courses</b>							
<b>S. No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>	<b>Cat. Code</b>
1	CS1011	Introduction to Computational Problem Solving	3	-	-	3	ESC
2	CS1012	Introduction to Computational Problem-Solving Lab	-	1	2	2	ESC
3	CS1031	Problem Solving through Computer Programming	3	-	-	3	ESC
4	CS1032	Problem Solving through Computer Programming Lab	-	1	2	2	ESC
5	CS1041	Introduction to Computer Programming	3	-	-	3	ESC
6	CS1042	Introduction to Computer Programming Lab	-	1	2	2	ESC
7	CS2101	Data Structures and Applications	3	-	-	3	ESC
8	CS2102	Data Structures and Applications Lab	-	-	2	1	ESC
9	CS2011	Discrete Event Simulation	1	-	-	1	ESC
10	CS2121	Basics of Algorithm Design	1	-	-	1	ESC

<b>CS1011</b>	<b>Introduction to Computational Problem Solving</b>	<b>ESC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--	------------	------------------	------------------

**Pre-requisites:** None

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Construct solutions to problems using computer with an understanding of the components of computing systems (apply)
<b>CO2</b>	Construct algorithms for mathematical and scientific problems (apply)
<b>CO3</b>	Construct modular programs using control structures and suitable data types (apply)
<b>CO4</b>	Compare alternate algorithmic approaches for problem solving and construct efficient algorithms (analyze)
<b>CO5</b>	Construct modular programs with an understanding of efficient memory access and usage (apply)

**Course Articulation Matrix:**

<b>CO \ PO</b>	<b>PO</b>											
	<b>PO1</b>	<b>PO2</b>	<b>PO3</b>	<b>PO4</b>	<b>PO5</b>	<b>PO6</b>	<b>PO7</b>	<b>PO8</b>	<b>PO9</b>	<b>PO10</b>	<b>PO11</b>	<b>PO12</b>
<b>CO1</b>	S	M	L									L
<b>CO2</b>	S	M	L									L
<b>CO3</b>	S	M	L									L
<b>CO4</b>	S	M	L									L
<b>CO5</b>	S	M	L									L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Fundamentals of Computers**, Historical perspective, Early computers, Components of a computers, Problems, Flowcharts, Memory, Variables, Values, Instructions, Programs.

**Problem solving techniques** – Algorithmic approach, characteristics of algorithm, Problem solving strategies: Top-down approach, Bottom-up approach, Time and space complexities of algorithms.

**Number systems and data representation**, Basics of C, Basic data types, Operators, Precedence and Associativity, Storage Classes, Dynamic Memory Allocation, Preprocessor Macros, Numbers, Digit separation, Reverse order, writing in words,

Development of Elementary School Arithmetic Testing System, Problems on Date and factorials, Solutions using flow of control constructs, Conditional statements - If-else, Switch-case constructs, Loops - while, do-while, for.

**Factoring Methods:** Finding the square root, Finding the smallest divisor of an integer, finding the greatest common divisor using Euclid's algorithm, Computing the prime factors of an integer, generating prime numbers, Raising a number to a large power, Computation of the nth Fibonacci number.

**Functions** – Modular approach for solving real time problems, user defined functions, library functions, parameter passing - call by value, call by reference, return values, Recursion, Introduction to pointers.

**Sorting and searching algorithms**, Large integer arithmetic, Single and Multi-Dimensional Arrays, passing arrays as parameters to functions.

Magic square and matrix operations using Pointers and Dynamic Arrays, Multidimensional Dynamic Arrays, String processing, File operations

**Structures and Unions** - Declaration, member variables, Problems on Complex numbers, Date, Time, Large Numbers.

**Reading list:**

1. R.G. Dromey, "How to solve it by Computer", Pearson, 2008.
2. Brian W.Kernighan, Dennis Ritchie, "The C Programming Language", 2<sup>nd</sup> edition, Person Education India, 2015
3. Behrauz, A. Forouzan, "Computer Science: A Structured Programming Approach Using C", 3<sup>rd</sup> edition, Cengage
4. Hanly J R & Koffman E.B, "Problem Solving and Programm design in C", 7th edition, Pearson Education
5. Randal E. Bryant, David R. O'Hallaron, "Computer Systems. A Programmer's Perspective", 2<sup>nd</sup> Edition, Prentice Hall
6. Ron white and Timothy Edward Downs, "How Computers work: The evolution of Technology" 10<sup>th</sup> Edition BPB Publications.

<b>CS1012</b>	<b>Introduction to Computational Problem-Solving Lab</b>	<b>ESC</b>	<b>0 – 1 – 2</b>	<b>2 Credits</b>
---------------	--	------------	------------------	------------------

**Pre-requisites:** None

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Construct, debug, test and run efficient programs by leveraging suitable flow of control constructs and syntactic units of the programming language.
<b>CO2</b>	Construct efficient programs by constructing and translating algorithms for solving problems using sorting, searching, selection and / or arithmetic computations.
<b>CO3</b>	Implement, refactor, test and debug functional programs in a shell-based run time environment.
<b>CO4</b>	Construct efficient programs by demonstrating problem-solving skills and out-of-the-box algorithmic thinking.

**Course Articulation Matrix:**

<b>CO \ PO</b>	<b>PO1</b>	<b>PO2</b>	<b>PO3</b>	<b>PO4</b>	<b>PO5</b>	<b>PO6</b>	<b>PO7</b>	<b>PO8</b>	<b>PO9</b>	<b>PO10</b>	<b>PO11</b>	<b>PO12</b>
<b>CO1</b>	S	M	L		S				M			L
<b>CO2</b>	S	M	L		S				M			L
<b>CO3</b>	S	M	L		S				M			L
<b>CO4</b>	S	M	L		S				M			L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Introduction to fundamentals of Ubuntu**, Operators, Storage classes, Dynamic memory allocation, Conditional statements, Iterative statements, Factoring methods, Functions, Pointers, Single and Multidimensional Arrays, Searching and Sorting algorithms, String operations, File operations, Structures and Union.

**Reading List:**

1. R.G. Dromey, "How to solve it by Computer", Pearson, 2008.
2. Brian W.Kernighan, Dennis Ritchie, "The C Programming Language", 2<sup>nd</sup> edition, Person Education India, 2015

3. Behrauz, A. Forouzan, "Computer Science: A Structured Programming Approach Using C", 3<sup>rd</sup> edition, Cengage
4. Hanly J R & Koffman E.B, "Problem Solving and Programm design in C", 7th edition, Pearson Education
5. Randal E. Bryant, David R. O'Hallaron, "Computer Systems. A Programmer's Perspective", 2<sup>nd</sup> Edition, Prentice Hall
6. Ron white and Timothy Edward Downs, "How Computers work: The evolution of Technology" 10<sup>th</sup> Edition BPB Publications.

<b>CS1031</b>	<b>Problem Solving through Computer Programming</b>	<b>ESC</b>	<b>3-0-0</b>	<b>3 Credits</b>
---------------	---	------------	--------------	------------------

**Pre-requisites:** None

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Apply problem-solving strategies such as top-down and bottom-up approaches to design simple algorithms and flowcharts for computational problems.
<b>CO2</b>	Construct algorithms for solving problems that requires solutions involving searching, sorting, selection and / or a numerical method as a sub-routine.
<b>CO3</b>	Analyze the suitability of different algorithmic design paradigms for solving problems with an understanding of the time and space complexities incurred.
<b>CO4</b>	Construct efficient modular programs for implementing algorithms by leveraging suitable control structures.
<b>CO5</b>	Construct efficient programs by selecting and using suitable in-built Data Structures and programming language features available.

**Course Articulation Matrix:**

CO \ PO	PO											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
<b>CO1</b>	S	M	L									L
<b>CO2</b>	S	M	L									L
<b>CO3</b>	S	M	L									L
<b>CO4</b>	S	M	L									L
<b>CO5</b>	S	M	L									L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Fundamentals of Computers**, Historical perspective, Early computers, Modern Computers, Hardware Components of a Computer, Data Representation in Computers, Introduction to Operating Systems, Software and Firmware, Problems, Flowcharts, Memory, Variables, Values, Instructions, Programs.

**Problem solving techniques** – Algorithmic approach, characteristics of algorithms, Problem solving strategies: Top-down approach, Bottom-up approach, Time and space complexities of algorithms, Algorithm Analysis.

**Basic Syntax in Python**, Data Types, Variables, Assignments, immutable variables, Types of Operators, Expressions, Comments, Boolean Logic, Logical Operators in Python. Conditional statements - If-else, Loops - while, for, Lazy Evaluation Inbuilt Data Structures and their operations in Python: List, Tuples and Dictionaries.

**Fundamental Algorithms**: Swapping variables, Problems involving summation of a series, Sine function computation, Base Conversion, generation of sequences like Fibonacci, Reversing the digits of an integer, Character to number conversion.

**Factoring Methods**: Finding the square root, Finding the smallest divisor of an integer, finding the greatest common divisor using Euclid’s algorithm, Computing the prime factors of an integer, generating prime numbers, Raising a number to a large power, Computation of the nth Fibonacci number.

**Functions** – Modular programming and benefits, user defined functions, library functions, parameter passing, Formal and Actual arguments, named arguments return values, Recursion.

**Sorting algorithms**: Bubble, Selection and Insertion sorts, Search algorithms: Linear and binary search

**String processing**: Algorithms for implementing String functions like Strlen, Strcpy, StrRev, Strcmp, Searching for a keyword or pattern in a text.

**File and Directory Handling**: Reading and Writing to/from a file, Formatted File creation and operations.

#### **Reading list:**

1. Kenneth Lambert, Fundamentals of Python: First Programs, Cengage Learning, 2019
2. R.G. Dromey, how to solve it by Computer, Pearson, 2008.
3. Brian W.Kernighan, Dennis Ritchie, “The C Programming Language”, 2<sup>nd</sup> edition, Person Education India, 2015
4. Hanly J R & Koffman E.B, “Problem Solving and Programm design in C”, 7th edition, Pearson Education
5. Randal E. Bryant, David R. O’Hallaron, “Computer Systems. A Programmer’s Perspective”, 2<sup>nd</sup> Edition, Prentice Hall

<b>CS1032</b>	<b>Problem Solving through Computer Programming Lab</b>	<b>ESC</b>	<b>0 – 1 – 2</b>	<b>2 Credits</b>
---------------	---	------------	------------------	------------------

**Pre-requisites:** None

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Construct, debug, test and run efficient programs by leveraging suitable flow of control constructs and syntactic units of the programming language.
<b>CO2</b>	Construct efficient programs by constructing and translating algorithms for solving problems using sorting, searching, selection and / or arithmetic computations.
<b>CO3</b>	Implement, refactor, test and debug functional programs in a shell-based run time environment.
<b>CO4</b>	Apply algorithmic thinking to break down complex problems into logical steps and implement efficient solutions through structured programming constructs.

**Course Articulation Matrix:**

CO \ PO	PO											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
<b>CO1</b>	S	M	L		S				M			L
<b>CO2</b>	S	M	L		S				M			L
<b>CO3</b>	S	M	L		S				M			L
<b>CO4</b>	S	M	L		S				M			L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Familiarization with Python installation**, basic syntax and running scripts in the shell, conditional control constructs, iterative constructs. (While, do-while, for), user defined functions and in-built function calls, Recursion, in-built data structures like List, Tuples and Dictionaries, String processing, Files and I/O.

**Implementation of Factoring methods**, sorting, searching and selection as sub-routines, simple 2D graphics.

Implementation of a capstone application to unify the concepts learnt in the course.

**Reading List:**

1. Kenneth Lambert, Fundamentals of Python: First Programs, Cengage Learning, 2019.
2. R.G. Dromey, how to solve it by Computer, Pearson, 2008.
3. Hanly J R & Koffman E.B, "Problem Solving and Programm design in C", 7th edition, Pearson Education
4. Randal E. Bryant, David R. O'Hallaron, "Computer Systems. A Programmer's Perspective", 2<sup>nd</sup> Edition, Prentice Hall
5. The Python Tutorial, Available at: <https://docs.python.org/3/tutorial/>.

<b>CS1041</b>	<b>Introduction to Computer Programming</b>	<b>ESC</b>	<b>3-0-0</b>	<b>3 Credits</b>
---------------	---	------------	--------------	------------------

**Pre-requisites:** None

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Construct flowcharts and write basic Python programs using variables, data types, expressions, conditional and loop control structures to solve computational problems.
<b>CO2</b>	Apply control structures to develop modular programs to solve mathematical problems
<b>CO3</b>	Design algorithms for solving simple mathematical problems including computing, searching and sorting
<b>CO4</b>	Demonstrate the use of Python's inbuilt data structures such as lists, tuples, and dictionaries to perform various data manipulation operations.
<b>CO5</b>	Implement string processing algorithms and file handling operations in Python, including the creation, reading, and writing of structured and unstructured data.
<b>CO6</b>	Utilize Python libraries such as pandas and turtle to create, analyze and visualize data using data frames and basic 2D graphics.

**Course Articulation Matrix:**

CO \ PO	PO											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
<b>CO1</b>	S	M	L									L
<b>CO2</b>	S	M	L									L
<b>CO3</b>	S	M	L									L
<b>CO4</b>	S	M	L									L
<b>CO5</b>	S	M	L									L
<b>CO6</b>	S	M	L									L

S: Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

**Fundamentals of Computers**, Historical perspective, Early computers, Modern Computers, Hardware Components of a Computer, Data Representation in Computers, Introduction to Operating Systems, Software and Firmware, Problems, Flowcharts, Memory, Variables, Values, Instructions, Programs.

**Basic Syntax in Python**, Data Types, Variables, Assignments, immutable variables, Types of Operators, Expressions, Comments, Boolean Logic, Logical Operators in Python. Conditional statements - If-else, Loops - while, for, Lazy Evaluation Inbuilt Data Structures and their operations in Python: List, Tuples and Dictionaries.

**Fundamental Algorithms**: Swapping variables, Problems involving summation of a series, Sine function computation, Base Conversion, generation of sequences like Fibonacci, Reversing the digits of an integer, Character to number conversion.

**Functions** – Modular programming and benefits, user defined functions, library functions, parameter passing, Formal and Actual arguments, named arguments return values, Recursion.

**Sorting algorithms**: Bubble, Selection and Insertion sorts, Search algorithms: Linear and binary search

**String processing**: Algorithms for implementing String functions like Strlen, Strcpy, StrRev, Strcmp, Searching for a keyword or pattern in a text.

**File and Directory Handling**: Reading and Writing to/from a file, Formatted File creation and operations. Introduction to pandas data frames, creation of data frames, querying data frames, merging data frames, reading and writing csv, excel files.

**Simple 2D Graphics**, drawing 2D objects using Turtle Graphics.

## **Reading list:**

1. Kenneth Lambert, Fundamentals of Python: First Programs, Cengage Learning, 2019
2. Wes McKinney and the Pandas Development Team. pandas: powerful Python data analysis toolkit Release 1.4.4
3. Ron white and Timothy Edward Downs, “How Computers work: The evolution of Technology” 10<sup>th</sup> Edition BPB Publications.

<b>CS1042</b>	<b>Introduction to Computer Programming Lab</b>	<b>ESC</b>	<b>0 – 1 – 2</b>	<b>2 Credits</b>
---------------	---	------------	------------------	------------------

**Pre-requisites:** None

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Design and test programs to solve mathematical problems.
<b>CO2</b>	Develop and test programs using control structures.
<b>CO3</b>	Construct modular programs using functions.
<b>CO4</b>	Construct programs to process strings and files.
<b>CO5</b>	Develop programs for visualizing data
<b>CO6</b>	Develop programs for data-wrangling using data frames

**Course Articulation Matrix:**

<b>PO</b> <b>CO</b>	<b>PO1</b>	<b>PO2</b>	<b>PO3</b>	<b>PO4</b>	<b>PO5</b>	<b>PO6</b>	<b>PO7</b>	<b>PO8</b>	<b>PO9</b>	<b>PO10</b>	<b>PO11</b>	<b>PO12</b>
<b>CO1</b>	S	M	L		S				M			L
<b>CO2</b>	S	M	L		S				M			L
<b>CO3</b>	S	M	L		S				M			L
<b>CO4</b>	S	M	L		S				M			L
<b>CO5</b>	S	M	L		S				M			L
<b>CO6</b>	S	M	L		S				M			L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Introduction to fundamentals of Ubuntu**, Operators, Conditional statements, Iterative statements, Functions, Pointers, Parameter passing mechanisms, Recursion, Single and Multidimensional Arrays, Searching and Sorting algorithms.

**Processing Strings**, Indexing Slicing, File operations: working with text files, structured files.

**Application of built-in data structures** like lists, dictionaries, tuples for data wrangling  
**Pandas Programming:** Creation, reading different formats, merging and querying data frames, aggregation, cross tabulation, pivoting.

**Plotting:** Line, Pie, Bar, Scatter Plots, Histogram, Plotting Graphs, Sub-Plots, Box Plot.

## Reading List:

1. Kenneth Lambert, Fundamentals of Python: First Programs, Cengage Learning, 2019
2. McKinney, Wes. "CS-108: Python for Data Analysis Data Wrangling with pandas, NumPy, and Jupyter." (2022). ISBN: 978-1-098-10403-0
3. Matplotlib: <https://matplotlib.org/stable/tutorials/>
4. Seaborn: <https://seaborn.pydata.org/tutorial.html>
5. Python – Data Visualization Series: <https://realpython.com/tutorials/data-viz/>

<b>CS2101</b>	<b>Data Structures and Applications</b>	<b>ESC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---	------------	------------------	------------------

**Pre-requisites:**

- i. Problem Solving through Computer Programming (CS1031)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Construct Abstract Data Types for modelling entities using appropriate data constructs and methods. (Apply)
<b>CO2</b>	Construct list-based data structures namely Stacks, Queues, Circular Queues and Linked Lists. (Apply)
<b>CO3</b>	Construct non-linear data structures namely Trees & graphs and set-based structures like disjoint sets. (Apply)
<b>CO4</b>	Construct suitable data structures and algorithms to facilitate searching, sorting and selection. (Apply)
<b>CO5</b>	Construct efficient algorithms for performing operations on data structures within a given time and /or space complexity. (Apply)
<b>CO6</b>	Assess the suitability of various data structures for solving a given problem with a comprehension of trade-offs in time and space complexities. (Analyze)

**Course Articulation Matrix:**

<b>CO</b>	<b>PO</b>											
	<b>PO1</b>	<b>PO2</b>	<b>PO3</b>	<b>PO4</b>	<b>PO5</b>	<b>PO6</b>	<b>PO7</b>	<b>PO8</b>	<b>PO9</b>	<b>PO10</b>	<b>PO11</b>	<b>PO12</b>
<b>CO1</b>	S	M	L		M							
<b>CO2</b>	S	M	L		M							
<b>CO3</b>	S	M	L		M							
<b>CO4</b>	S	M	L		M							
<b>CO5</b>	S	M	L		M							
<b>CO6</b>	S	S	S	M	M							

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Introduction to Data Structures**, Algorithm Analysis and Examples based on Asymptotic Notations, Abstract Data Types (ADTs), Stacks, Queues, Circular Queues and Linked List (Singly Linked, Doubly Linked and Circular). Applications of Stacks, Queues and Linked Lists.

**Trees:** Representation of Trees, Binary Trees, Binary Search Trees, Applications of Trees.

**Priority Queues,** Binary Heap and applications

**Hash Tables and Operations,** Collision Resolution: Open Addressing and Chaining, Applications of Hash Tables.

**Graphs:** Representation of Graphs, Graph Traversal Techniques, Minimum Cost Spanning Trees: Prim's and Kruskal's Algorithms, Shortest Path Algorithms: Dijkstra's Algorithm and Floyd-Warshall Algorithm. Applications of Graphs.

**Sorting Algorithms:** Merge Sort, Heap Sort, Quick Sort and Counting Sort.

### **Reading List:**

1. Michael T. Goodrich, R. Tamassia, and Mount, Data Structures and Algorithms in C++, Second Edition, John Wiley and Sons, 2011.
2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Fourth Edition, Pearson Education. Ltd., 2014.
3. Adam Drozdek, Thomson, Data structures and algorithms in C++, Fourth Edition, Cengage, 2013.
4. Richard F. Gilberg, Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C++, Second Edition, Thomson Learning, 2004.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.

<b>CS2102</b>	<b>Data Structures and Applications Lab</b>	<b>ESC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	---	------------	------------------	-----------------

**Pre-requisites:**

- i. Problem Solving through Computer Programming (CS1031)
- ii. Problem Solving through Computer Programming Lab (CS1032)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Construct solutions for problems using linear data structures such as Linked List, Stacks and Queues. (Apply)
<b>CO2</b>	Construct solutions for problems using non-linear Data Structures such as Trees and Graphs. (Apply)
<b>CO3</b>	Implement solutions for problems that requires sorting and searching as a sub-routine. (Apply)
<b>CO4</b>	Analyze, evaluate and choose appropriate data structures and algorithms for a specific application. (Analyze)
<b>CO5</b>	Analyze algorithms with respect to their time and space complexities. (Analyze)

**Course Articulation Matrix:**

<b>CO \ PO</b>	<b>PO1</b>	<b>PO2</b>	<b>PO3</b>	<b>PO4</b>	<b>PO5</b>	<b>PO6</b>	<b>PO7</b>	<b>PO8</b>	<b>PO9</b>	<b>PO10</b>	<b>PO11</b>	<b>PO12</b>
<b>CO1</b>	S	M	L		S			S	S	M		
<b>CO2</b>	S	M	L		S			S	S	M		
<b>CO3</b>	S	M	L		S			S	S	M		
<b>CO4</b>	S	M	M	L	S			S	S	M		
<b>CO5</b>	S	M	M	L	S			S	S	M		

S: Strong correlation, M: Medium correlation, L: Low correlation

**Tentative List of Experiments:**

1. Implementation of Stack and Queue using Arrays.
2. Implementation of Stack-based applications like postfix expression evaluation and infix to postfix conversion.
3. Implementation of Queue and Circular Queue.

4. Implementation of Single Linked List, Doubly Linked List and Circular Linked List.
5. Implementation of Stack and Queue using Linked List.
6. Representative problems with solutions involving Stack, Queue and Linked List.
7. Implementation of Binary Search Tree.
8. Implementation of BST traversals in recursive and non-recursive ways.
9. Implementation of AVL Tree.
10. Implementation of Priority Queue.
11. Implementation of Dictionaries using open and closed addressing schemes.
12. Implementation of Trie for fast text matching.
13. Implementation of Quick, Merge, Counting, Radix and Bucket sorts.
14. Implementation of Graphs and Depth First & Breadth First Traversals.
15. Mini project involving design, memory organization, implementation and complexity analysis of data structures and their associated operations.
16. Case study/Mini project

**Reading List:**

1. Michael T. Goodrich, R. Tamassia, and Mount, Data Structures and Algorithms in C++, Second Edition, John Wiley and Sons, 2011.
2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Fourth Edition, Pearson Education. Ltd., 2014.
3. Adam Drozdek, Thomson, Data structures and algorithms in C++, Fourth Edition, Cengage, 2013.
4. Richard F. Gilberg, Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C++, Second Edition, Thomson Learning, 2004.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.

<b>CS2011</b>	<b>Discrete Event Simulation</b>	<b>DEC</b>	<b>1 – 0 – 0</b>	<b>1 Credits</b>
---------------	----------------------------------	------------	------------------	------------------

This course introduces students to the underlying theory, motivation and tools and techniques to construct discrete event simulations to model and analyze the performance of a computing system or its component.

**Pre-requisites:**

- i. Introduction to Algorithmic Thinking and Programming (CS1101)
- ii. Data Structures and Algorithms (CS201)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Construct discrete event simulation models for characterizing a computing system or its component. (Apply)
<b>CO2</b>	Verify and validate discrete event simulation models (Apply)
<b>CO3</b>	Implement discrete event simulation models using a suitable simulation software and analyze the simulation results using suitable performance metrics. (Analyze)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	L	L									S	S	S
<b>CO2</b>	S	M	L	L									S	S	S
<b>CO3</b>	S	S	M	M	S					S		S	S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Introduction to Simulation:** Scenarios requiring simulations, Merits and demerits of Simulation, Applications, System – environment, components and models, Types of models, Discrete event system simulation, Examples involving queuing and inventory systems.

**Simulation Modelling** – Types of Simulations: Monte-Carlo, Trace-driven, Discrete Event Simulations: Event Set Algorithms, Random number generation, Random variate

generation, Testing random numbers, Analysis of Simulation Results.

**Selection of Simulation Software**, Examples such as SIMUL8, GPSS, Arena, Network Simulator, Quantum Simulators.

**Reading List:**

1. Jerry Banks, Hon S Carson, Barry L Nelson, David M Nicol, "Discrete Event Simulation 5<sup>th</sup> Edition, Pearson, 2010.
2. Raj Jain, The Art of Computer Systems Performance Analysis, Wiley, 2008.

<b>CS2121</b>	<b>Basics of Algorithm Design</b>	<b>ESC</b>	<b>1-0-0</b>	<b>1 Credit</b>
---------------	-----------------------------------	------------	--------------	-----------------

**Prerequisites:** None

**Course Outcomes:** After completion of the course, students will be able to:

<b>CO1</b>	Analyze the efficiency of algorithms using asymptotic notations and solve recurrence relations through substitution, recursion-tree, and master theorem methods.
<b>CO2</b>	Design and evaluate various sorting and selection algorithms, including heapsort, quicksort, radix sort, and linear-time selection techniques.
<b>CO3</b>	Apply advanced algorithmic concepts for polynomial computations and demonstrate an understanding of NP-completeness and problem reductions.

### Detailed Syllabus:

**FOUNDATIONS:** Growth of Functions: Asymptotic notation, Standard notations and common functions, Divide-and-Conquer: The maximum-subarray problem, Strassen's algorithm for matrix multiplication, the substitution method for solving recurrences, the recursion-tree method for solving recurrences, the master method for solving recurrences, Proof of the master theorem.

**SORTING AND ORDER STATISTICS:** Heapsort: Heaps, Maintaining the heap property, building a heap, the heapsort algorithm, Quicksort: Description of quicksort, Performance of quicksort, A randomized version of quicksort, Analysis of quicksort, Sorting in Linear Time: Lower bounds for sorting, counting sort, Radix sort, Bucket sort, Medians and Order Statistics: Minimum and maximum, Selection in expected linear time, Selection in worst-case linear time.

**SELECTED TOPICS: Polynomials and the FFT:** Representing polynomials, The DFT and FFT, Efficient FFT implementations, NP-Completeness: Polynomial time, Polynomial-time verification, NP-completeness and reducibility, NP-completeness proofs, NP-complete problems.

### Reading List:

1. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to algorithms. MIT press, 2022.
2. Michael T. Goodrich, R. Tamassia, and Mount, Data Structures and Algorithms in C++, Second Edition, John Wiley and Sons, 2011.

## I Year Syllabus

<b>MA1011</b>	<b>Differential and Integral Calculus</b>	<b>BSC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---	------------	------------------	------------------

**Pre-requisites:** None.

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Assess the behavior of functions and identify their limits.
<b>CO2</b>	Apply derivative calculation techniques to compute the derivatives of different types of functions.
<b>CO3</b>	Apply integration techniques to solve integrals.
<b>CO4</b>	Apply integrals to analyze, model and solve various geometric and physical problems.
<b>CO5</b>	Apply standard theorems in calculus to analyze and solve complex engineering problems.

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M	L									S	M	M
<b>CO2</b>	S	M	M	L									S	M	M
<b>CO3</b>	S	M	M	L									S	M	M
<b>CO4</b>	S	S	M	L									S	M	M
<b>CO5</b>	S	S	M	L									S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Univariate Calculus:**

**Real number system:** Completeness axiom, density of rationals (irrationals) in real number field.

**Sequences:** Limits and convergence of sequences, Sandwich theorem, Monotone sequences, Subsequence, Bolzano-Weierstrass Theorem, Cauchy Convergence Criterion.

Limits, continuity, and differentiability of functions: Intermediate value theorem, extreme value theorem, chain rule, Rolle's theorem, mean value theorem, Taylor's theorem, maxima and minima.

**Integrability of functions:** Riemann integrals, Fundamental theorem of calculus, improper integrals, applications to area and volume.

Series: Convergence of a series, power series.

### **Multivariate Calculus:**

Review of vector algebra: Euclidean space

Limits, continuity, partial derivatives, gradient and directional derivatives, (total) derivative, chain rule, maxima and minima, Lagrange multipliers.

Integrability: Double and triple integrals, Fubini's theorem, Jacobians and change of variables formula.

Parametrization of curves and surfaces, vector fields, line and surface integrals.

Divergence and curl, Theorems of Green, Gauss, and Stokes.

### **Reading List:**

1. William G. McCallum, Deborah Hughes-Hallett, Andrew M. Gleason, Calculus: Single and Multivariable, 8th Edition Second Edition, 8th Edition, Wiley, 2020.
2. James Stewart, Essential Calculus, 2nd edition, Brooks/Cole, 2012.
3. T. M. Apostol, Calculus, Volumes 1 and 2 (2nd Edition), Wiley Eastern, 1980.
4. G. B. Thomas and R. L. Finney, Calculus and Analytic Geometry, 9th Edition, Pearson, 2010.

PH1011	ENGINEERING PHYSICS	2-0-0	Credits 2
--------	---------------------	-------	--------------

**Course Outcomes: At the end of the course, the student will be able to**

CO1	Solve engineering problems using the concepts of wave and particle nature of radiant energy
CO2	Understand the use of lasers as light sources for low and high energy applications
CO3	Construct quantum mechanical model to explain the behaviour of a system at microscopic level
CO4	Apply the concepts of energy harvesting and understand the mechanisms of photo-voltaic cells
CO5	Understand the nature and characterization of magnetic and nano-materials for applications

**Course Articulation Matrix**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	M	S	M										L		M
CO2	M	S											L		M
CO3		M											L		M
CO4	M	M	L						M				L		M
CO4	M	M	L										L		M

**Detailed Syllabus:**

### UNIT- I WAVE OPTICS

**Interference:** Superposition principle, coherence of light, methods to produce coherent light: division of amplitude and wavefront, Young's double slit and Newton's rings experiment: concept, working principle, and applications.

**Diffraction:** Introduction to Fresnel and Fraunhofer diffraction, Fraunhofer's single-slit diffraction, double-slit diffraction and diffraction grating (quantitative), and resolving power of a grating.

**Polarization:** Introduction to polarization, Types of polarization, Methods to produce polarization: Reflection, refraction, scattering, selective absorption and double refraction.

## **UNIT - II LASERS**

Basic theory of LASER, Einstein's coefficients and their relations, concept of population inversion, components of lasers, three and four level lasing systems, construction and working principle of various types of lasers: Nd-YAG, Helium-Neon and semiconductor lasers and their applications.

## **UNIT - III QUANTUM PHYSICS**

Origin of quantum theory and related experiments: Black-Body radiation and photo-electric effect. Heisenberg's uncertainty principle, de- Broglie's wave concept, wave function, and its properties, operators, Schrödinger's time-dependent and time-independent equations (Quantitative), particle in one-dimensional, infinite potential well, quantum tunneling phenomena and their applications in alpha decay, and scanning tunnelling microscopy (STM). Introduction to Quantum Technology (Q-switching, interaction of radiation with matter).

## **UNIT - IV PHOTOVOLTAICS**

Introduction to semiconductors, Solar spectrum, photovoltaic (PV) effect, materials, structure and working principle, I-V characteristics, power conversion efficiency, quantum efficiency, emerging PV technologies, and applications.

## **UNIT - V MAGNETIC AND NANO MATERIALS**

### **Magnetic Materials:**

Introduction to Weiss theory of ferromagnetism, concepts of magnetic domains, spontaneous magnetization, Curie transition, hard and soft magnetic materials and their applications.

### **Nanomaterials:**

Introduction, classification, and properties of nanomaterials, various methods of synthesizing nanomaterials: top-down (ball milling) and bottom-up (sol-gel) approaches.

### **Textbooks:**

1. A Textbook of Engineering Physics, Revised Edition, M. N. Avadhanulu, P. G. Kshirsagar, S. Chand and Company (2014).
2. Concepts of Modern Physics, 7<sup>th</sup> Edition, Beiser A., Mc. Graw Hill Publishers (2017).
3. Optics, Ajoy Ghatak, 7<sup>th</sup> Edition, Tata Mc Graw Hill (2020).
4. Lasers- Fundamentals and Applications, Ajoy Ghatak and K. Thyagarajan, 2<sup>nd</sup> Edition, Laxmi Publications (2019).

### **Reference Books:**

1. Materials Science and Engineering: An Introduction (Tenth edition), William D. Callister, John Wiley & Sons (2018).
2. Introduction to Solid State Physics, 8<sup>th</sup> Edition, Charles Kittel, Wiley Publishers (2012).

<b>Course Code:</b> PH1012	<b>Engineering Physics Laboratory</b>	<b>L-T-P:</b> 0-0-2	<b>Credits</b> 1
-------------------------------	---------------------------------------	------------------------	---------------------

**Pre-requisites:** None

**Course Outcomes:** At the end of the course, the students will be able to

<b>CO1</b>	Understand the basic properties of light by performing experiments on interference, diffraction and polarization.
<b>CO2</b>	Acquire the experimental knowledge by performing the experiment using light-emitting diode (LED) for energy conversion applications.
<b>CO3</b>	Understand the nature and characteristics of ferromagnetic and dielectric materials for memory device and sensor applications.
<b>CO4</b>	Apply the knowledge of Solar/ PV cells for choice of materials in efficient alternate
<b>CO5</b>	Apply the concepts of wave propagation through optical fibres and communication

#### Course Articulation Matrix

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	M	S	M										L		M
<b>CO2</b>	M	S											L		M
<b>CO3</b>		M											L		M
<b>CO4</b>	M	M	L						M				L		M

**List of experiments (any six experiments from the following list):**

Exposure to Virtual Lab (any two of the following):

S. No.	Name of the experiment
1	Determination of radius of curvature of plano-convex lens using Newton's ring experiment.
2	Determination of the width of narrow-slit by diffraction method
3	Determination of wavelength of spectral lines of Mercury light by normal incidence method using diffraction grating
4	Determination of Planck's constant using light emitting diode
5	Study the B-H loop hysteresis and find the coercivity and retentivity of magnetic materials
6	Studying current-voltage characteristics of a photovoltaic material using solar cell
7	Determination of wavelength of diode laser using diffraction by metal scale
8	Determination of dielectric constant of various dielectric materials
9	Determination of numerical aperture of an optical fibre
10	Determination of specific rotation of an optically active material-using Laurent's half-shade polarimeter

1. B-H Loop tracer
2. Planck's Constant
3. Numerical aperture of Optical Fibre
4. Newton's ring

**References:**

1. Physics Laboratory Manual, 2024, Department of Physics, School of Sciences, National Institute of Technology Andhra Pradesh, Tadepalligudem.
2. Shukla R K, Srivastava A, 2011, Practical Physics, New Age International Pvt. Ltd., New Delhi.
3. Arora, CL, 2012, B.Sc. Practical Physics, S Chand and Company Ltd., New Delhi.
4. A Textbook of Engineering Physics, M. N. Avadhanulu, P. G. Kshirsagar, S. Chand and Company (2015).
5. A Course of Experiments with He-Ne Lasers by R.S. Sirohi, New Age International (P) Ltd. (2009)

<b>EC1531</b>	<b>Digital Logic Design</b>	<b>ESC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-----------------------------	------------	------------------	------------------

**Pre-requisites:** None.

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Design of combinational and sequential logic circuits and develop Verilog model
<b>CO2</b>	Understand characteristics of the TTL/CMOS logic families and realize Boolean equation using CMOS logic
<b>CO3</b>	Understand the concepts of Combinational and Sequential Circuits
<b>CO4</b>	Understand SRAM/DRAM organization and periphery circuitry, operation of SRAM cell, DRAM cell, DDR2/DDR4 and FPGA/CPLD

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	M	S	M										L		M
<b>CO2</b>	M	S											L		M
<b>CO3</b>		M											L		M
<b>CO4</b>	M	M	L						M				L		M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Number Systems and Codes:** Representation of unsigned and signed integers, Floating Point representation of real numbers, Laws of Boolean Algebra, Theorems of Boolean Algebra, Realization of functions using logic gates, Canonical forms of Boolean Functions, Minimization of Functions using Karnaugh Maps.

**Combinational circuit design:** Design with basic logic gates, comparators, data selectors, priority encoders, decoders, full adder, serial binary adder, parallel binary adders-ripple-carry adder, carry-look ahead adder; Parallel prefix adders- Carry select Adder, Conditional sumadder, Kogge-stone Adder, Brent- kung adder, Verilog models.

**Sequential circuit design:** Memory elements and their excitation functions SR, JK, T, and D latches and flip-flops, master slave JK flip-flop, edge-triggered flip-flop, synchronous and asynchronous counters, finite-state machine, sequence detector, minimization and transformation of sequential machines, Registers, Verilog models

**Testing of Combinational circuits:** Fault models, structural testing: path sensitization  
Logic families: TTL and CMOS Logic circuits, Transfer characteristics, fan-in, fan-out, noise margin, rise time and fall time analysis, realization of Boolean equations using CMOS logic

**Memory:** Types of memories, MOS SRAM cells, DRAM, SDRAM, ALL DDRx, organization of a SRAM, Organization of SDRAM, Periphery circuitry of Memory, Flash memory, SD card.

**Reading List:**

1. Morris. M. Mano, Michael D. Ciletti, Digital Design. Fourth Edition, Prentice-Hall India. 2008.
2. William J. Dally and John W. Poulton, Digital Systems Engineering, Cambridge University Press, 2008.
3. Schilling, Herbert Taub and Donald, Digital Integrated Electronics, Tata McGraw-Hill, 2008.
4. Digital Design: With an Introduction to the Verilog HDL, VHDL, and System Verilog, 6th edition.
5. Jayaram Bhasker, Verilog Primer, Prentice-Hall India, 1998,3rd edition.
6. Thomas L. Floyd, Digital Fundamentals, 11th Edition, Pearson 2015
7. Sameer Palnitkar, Verilog HDL: A guide to digital Design and Synthesis, Pearson, 2003, 2nd edition.
8. John F Wakerly, Digital Design Principles and Practices, Prentice Hall India, 2001, 3rd Edition.
9. Franklin P. Processor, David E. Winkel, The Art of Digital Design: An Introduction to Top-Down Design, PTR Prentice Hall, 1987, 2nd Edition.

<b>CS1021</b>	<b>Computer Organization and Architecture</b>	<b>PCC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---	------------	------------------	------------------

**Pre-requisites:**

- i. Digital Logic Design (EC1531)

**Course Outcomes:** At the end of the course, students will be able to:

<b>CO1</b>	Construct assembly language programs to perform the specified task with a comprehension of addressing modes. (Apply)
<b>CO2</b>	Analyze the speed and cost of memory access based on availability of data in cache memory, main memory and secondary memory. (Apply)
<b>CO3</b>	Assess the performance of data transfer to and from the peripherals and identify the suitable mode of transfer. (Apply)
<b>CO4</b>	Design Arithmetic Logic Unit components and microprogrammed control units satisfying the specified constraints. (Apply)
<b>CO5</b>	Design pipelines for high performance considering possible hazards, data paths and control paths. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M	L						L			S	S	S
<b>CO2</b>	S	M	L	L						L			S	S	S
<b>CO3</b>	S	M	M	L						L			S	S	S
<b>CO4</b>	S	M	M	L						L			S	S	S
<b>CO5</b>	S	M	M	L						L			S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

## Detailed Syllabus:

**Basic Structures of Computers:** Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Software, Performance, Multiprocessors and Multi-computers, Historical Perspective.

**Machine instructions and Programs:** Numbers, Arithmetic Operations and Characters, Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Basic Input Output Operations, Stacks and Queues, Subroutines, Additional Instructions, Example Programs, Encoding of Machine Instructions.

Registers and addressing, IA-32 Instructions, IA-32 Assembly Language, Program Flow Control, Logic and Shift/Rotate Instructions, I/O Operations, Subroutines, Other Instructions, Program Examples.

**Input/output Organization:** Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces.

**The Memory System:** Some Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed Size and Cost, Cache Memories: mapping functions, cache performance and replacement algorithms, levels and locality, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage.

**Arithmetic:** Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-Operand Multiplication, Fast Multiplication, Integer Division, Floating Point Numbers and Operations, Implementing Floating Point Operations.

**Basic Processing Unit:** Some Fundamental Concepts, Execution of a Complete Instruction, Multiple-Bus Organization, Hardwired Control, Microprogrammed Control.

**Pipelining:** Basic Concepts, Data Hazards, Instruction Hazards, Influence on Instruction Sets, Data Path and Control Considerations, Super Scalar Operation, UltraSPARC 2 Example, Performance Consideration.

Introduction to superscalar architectures, VLIW machines.

## Reading List:

1. Carl Hamacher, Computer Organization, 5th Edition, McGraw Hill Publishers, 2002.
2. William Stallings, Computer Organization and Architecture Designing for Performance, 8th Edition, Pearson Education, 2010.
3. J.L. Hennessy and D.A. Patterson, Computer Architecture: A Quantitative Approach, 5th Edition, Morgan Kauffmann Publishers, 2012.
4. Morris Mano, Computer System Architecture, 3rd Edition, PHI, 2017.
5. Hennessy, J. L., and D. A. Patterson, Computer Architecture: A Quantitative Approach, 3rd edition. San Mateo, CA: Morgan Kaufman, 2002.

<b>MA1021</b>	<b>Matrices and Differential Equations</b>	<b>BSC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--	------------	------------------	------------------

**Pre-requisites:** None.

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Prove theorems and solve problems in linear algebra with a comprehension of matrices, vector algebra and linear equations. (Apply)
<b>CO2</b>	Infer eigen values, eigen vectors and rank nullity of matrices. (Apply)
<b>CO3</b>	Decompose matrices using suitable matrix factorization techniques. (Apply)
<b>CO4</b>	Solve first order and second order differential equations. (Apply)
<b>CO5</b>	Solve complex engineering problems using Laplace and Fourier transforms. (Analyze)

### Course Articulation Matrix

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M	L						L			S	S	S
<b>CO2</b>	S	M	L	L						L			S	S	S
<b>CO3</b>	S	M	M	L						L			S	S	S
<b>CO4</b>	S	M	M	L						L			S	S	S
<b>CO5</b>	S	M	M	L						L			S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

### Detailed Syllabus:

#### Matrix Theory:

Linear dependence and independence of vectors; Rank of a matrix; Consistency of the system of linear equations; Solving a system of linear equations by using inverse matrix method; Eigenvalues and eigenvectors of a matrix; Caley-Hamilton theorem and its applications; Reduction to diagonal form; Reduction of a quadratic form to canonical form - orthogonal transformation; Upper and Lower Triangulation; Singular Value Decomposition; Properties of complex matrices - Hermitian, skew-Hermitian and Unitary matrices.

**Ordinary Differential Equations:**

Higher order linear differential equations with constant coefficients - homogeneous and non-homogeneous; Euler and Cauchy's differential equations; Method of variation of parameters; System of linear differential equations; applications in physical problems such as forced oscillations and electric circuits.

**Transforms:**

Laplace transforms; inverse Laplace transforms; Properties of Laplace transforms; Laplace transforms of unit step function, impulse function, periodic function; Convolution theorem, solving certain initial value problems, Solving system of linear differential equations

Fourier Series, The Fast Fourier Transform.

**Reading List:**

1. E. Kreyszig, Advanced Engineering Mathematics, Eighth Edition, John Wiley and Sons, 2015.
2. B. S. Grewal, Higher Engineering Mathematics, Khanna Publications, 2015.
3. R. K. Jain and S. R. K. Iyengar, Advanced Engineering Mathematics, Fifth Edition, Narosa Publishing House, 2016.
4. G. Strang, Linear Algebra and Its Applications, 4th Edition, Brooks/Cole India, 2006.
5. T. M. Apostol, Calculus, Volume 2 (2nd Edition), Wiley Eastern, 1980

<b>MA1071</b>	<b>Discrete Mathematics</b>	<b>BSC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-----------------------------	------------	------------------	------------------

**Pre-requisites:** None.

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Construct mathematical proofs using mathematical logic and induction. (Apply)
<b>CO2</b>	Apply logical notations to define and reason mathematically about the constructs used in computing. (Apply)
<b>CO3</b>	Apply techniques for counting the occurrences of discrete events including permutations, combinations with or without repetitions. (Apply)
<b>CO4</b>	Solve problems using graph representations. (Apply)
<b>CO5</b>	Formulate and solve recurrence relations. (Apply)
<b>CO6</b>	Identify the algebraic structures and verify their properties. (Analyze)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M	L						L			S	S	S
<b>CO2</b>	S	M	L	L						L			S	S	S
<b>CO3</b>	S	M	M	L						L			S	S	S
<b>CO4</b>	S	M	M	L						L			S	S	S
<b>CO5</b>	S	M	M	L						L			S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**A review of Sets and Relations:** Sets, Operations on Sets, Venn Diagrams, Multi Sets, Binary Relations, Equivalence Relations, Ordering Relations, Operations on Relations, Partial Orders.

**A review of Functions:** Definition and Introduction, Composition of Functions, Inverse Functions, Binary and n-ary Operations.

**Mathematical Logic and Induction:** Statements and Notation, Connectives, Quantified Propositions, Logical Inferences, Methods of Proof of an Implication, First Order Logic

and other Methods of Proof, Rules of Inference for Quantified Propositions, Proof by Mathematical Induction.

**Elementary Combinatorics:** Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficients, The Binomial and Multinomial Theorems, The Principle of Inclusion- Exclusion.

**Recurrence Relations:** Generating Functions of Sequences, Calculating Coefficients of Generating Functions, Recurrence Relations, Solving Recurrence Relations by Substitution and Generating Functions, The Method of Characteristic Roots, Solutions of Inhomogeneous Recurrence Relations.

**Lattices as Partially Ordered Sets:** Definition and Examples, Properties of Lattices, Lattices as Algebraic Systems, Sublattices, Direct Product, Homomorphism, Some Special lattices.

**Graphs:** Basic Definitions Undirected and Directed Graphs, Paths, Representation of Graphs, Reachability, Connected Components, Examples of Special graphs, Graph Isomorphism, Planar Graphs, Euler's Formula, Euler Circuits, Hamiltonian Graphs, Chromatic Number of a Graph, The Four-Color Problem, Graph Traversals, Applications of Graphs.

**Algebraic Structures:** Groups, Semigroups, Monoids, Rings and Fields.

### Reading List:

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, 7<sup>th</sup> Edition, Tata McGraw Hill Publishers, 2007.
2. K.V. Iyer, Karthick Seshadri, Discrete Mathematics with Applications, First Edition, Cengage, 2022.
3. Joe L. Mott, Abraham Kandel, Theodore P. Baker, Discrete Mathematics for Computer Scientists and Mathematicians, Second Edition, PHI, 2001.
4. Norman L Biggs, Discrete Mathematics, 2<sup>nd</sup> Edition, Indian Edition published by Oxford University Press.
5. Tremblay J. P. and Manohar R., Discrete Mathematical Structures, MGH, 1997.

<b>CS1051</b>	<b>Data Structures and Algorithms</b>	<b>PCC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---------------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Introduction to Computational Problem Solving (CS1011)
- ii. Introduction to Computational Problem-Solving lab (CS1012)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Construct Abstract Data Types for modelling entities using appropriate data constructs and methods. (Apply)
<b>CO2</b>	Construct list-based data structures namely Stacks, Queues, Circular Queues and Linked Lists. (Apply)
<b>CO3</b>	Construct non-linear data structures namely Trees & graphs and set-based structures like disjoint sets. (Apply)
<b>CO4</b>	Construct suitable data structures and algorithms to facilitate searching, sorting and selection. (Apply)
<b>CO5</b>	Construct efficient algorithms for performing operations on data structures within a given time and /or space complexity. (Apply)
<b>CO6</b>	Assess the suitability of various data structures for solving a given problem with a comprehension of trade-offs in time and space complexities. (Analyze)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	L										S	S	M
<b>CO2</b>	S	M	L										S	S	M
<b>CO3</b>	S	M	L										S	S	M
<b>CO4</b>	S	M	L										S	S	M
<b>CO5</b>	S	M	L										S	S	M
<b>CO6</b>	S	S	S	M									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

Abstract Data Types (ADT), Arrays and Records, Memory Layout and implementation of Stack, Queue, Linked List (Single, Double and Circular), Applications of Stack, Queue and Linked Lists. Implementation of variants of the Stack, Queue and Linked List such as Two-way Stack, Circular Queue and Doubly Circular Linked Lists. Implementation of Stack and Queue using Linked List.

Memory Layout and Implementation of operations on Trees: Binary Trees, Binary Search Trees, AVL trees, B-Trees, Splay Trees and Tries.

Hashing: Open and Closed addressed tables, Design of good hash functions, Analysis of the efficiency of search operations, Bloom Filters.

Priority Queues: Memory layout, implementation and applications of Heap and its variants namely Skew Heaps, Leftist Heaps, Fibonacci Heaps and Binomial Queues.

Miscellaneous Data Structures: Skip Lists, Disjoint sets and k-d Trees.

Graphs: Memory representations, Depth First and Breadth First Traversals and representative applications like Topological Sorting, Finding Connected Components and Social Network Analysis.

Sorting: Representative Internal Sorting Algorithms like Merge Sort, Quick Sort, Counting Sort, Radix and Bucket Sorts. Analysis of efficiency of sorting, Notion of Stability in Sorting, External Sorting Algorithms.

Amortized Analysis of Data Structures: Accounting and Potential methods.

## **Reading List:**

1. Michael T. Goodrich, R. Tamassia, and Mount, Data Structures and Algorithms in C++, Second Edition, John Wiley and Sons, 2011.
2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Fourth Edition, Pearson Education. Ltd., 2014.
3. Adam Drozdek, Thomson, Data structures and algorithms in C++, Fourth Edition, Cengage, 2013.
4. Richard F. Gilberg, Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C++, Second Edition, Thomson Learning, 2004.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.

<b>CS1052</b>	<b>Data Structures and Algorithms lab</b>	<b>PCC</b>	<b>0 – 1 – 2</b>	<b>2 Credits</b>
---------------	---	------------	------------------	------------------

**Pre-requisites:**

- i. Problem Solving through Computer Programming (CS1011)
- ii. Problem Solving through Computer Programming lab (CS1012)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Implement Abstract Data Types for modeling entities using appropriate data constructs and methods. (Apply)
<b>CO2</b>	Implement list-based data structures namely Stacks, Queues, Circular Queues and Linked Lists. (Apply)
<b>CO3</b>	Implement non-linear data structures namely Trees & graphs and set-based structures like disjoint sets. (Apply)
<b>CO4</b>	Implement suitable efficient data structures and algorithms to facilitate searching, sorting and selection. (Apply)
<b>CO5</b>	Implement efficient algorithms for performing operations on data structures within a given time and/or space complexity. (Apply)
<b>CO6</b>	Implement solutions after assessing the suitability of different data structures for solving a given problem with a comprehension of trade-offs in time and space complexities by demonstrating problem-solving and programming skills. (Analyze)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M						S	M	L		S	S	M
<b>CO2</b>	S	M	M		S				S	M	L		S	S	M
<b>CO3</b>	S	M	M		S				S	M	L		S	S	M
<b>CO4</b>	S	M	M		S				S	M	L		S	S	M
<b>CO5</b>	S	M	M		S				S	M	L		S	S	M
<b>CO6</b>	S	S	S	M	L				S	M	L		S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

### **Tentative List of Experiments:**

1. Implementation of Stack and Queue using Arrays.
2. Implementation of Stack-based applications like postfix expression evaluation and infix to postfix conversion.
3. Implementation of Queue and Circular Queue.
4. Implementation of Single Linked List, Doubly Linked List and Circular Linked List.
5. Implementation of Stack and Queue using Linked List.
6. Representative problems with solutions involving Stack, Queue and Linked List.
7. Implementation of Binary Search Tree.
8. Implementation of BST traversals in recursive and non-recursive ways.
9. Implementation of AVL Tree.
10. Implementation of Priority Queue.
11. Implementation of Dictionaries using open and closed addressing schemes.
12. Implementation of Trie for fast text matching.
13. Implementation of Quick, Merge, Counting, Radix and Bucket sorts.
14. Implementation of Graphs and Depth First & Breadth First Traversals.
15. Mini project involving design, memory organization, implementation and complexity analysis of data structures and their associated operations.
16. Case study/Mini project.

### **Reading List:**

1. Michael T. Goodrich, R. Tamassia, and Mount, Data Structures and Algorithms in C++, Second Edition, John Wiley and Sons, 2011.
2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Fourth Edition, Pearson Education. Ltd., 2014.
3. Adam Drozdek, Thomson, Data structures and algorithms in C++, Fourth Edition, Cengage, 2013.
4. Richard F. Gilberg, Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C++, Second Edition, Thomson Learning, 2004.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.

<b>CS1061</b>	<b>Object Oriented Design and Programming</b>	<b>PCC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---	------------	------------------	------------------

**Pre-requisites:**

- i. Problem Solving through Computer Programming (CS1011)
- ii. Problem Solving through Computer Programming lab (CS1012)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Construct programs using Object Oriented Design principles like encapsulation, abstraction, polymorphism, inheritance and typing. (Apply)
<b>CO2</b>	Develop applications with handlers for user-defined exceptions, according to the given requirements. (Apply)
<b>CO3</b>	Construct efficient multi-threaded programs with synchronization constructs. (Apply)
<b>CO4</b>	Develop interactive GUI applications with event handling that provide rich user experience. (Apply)
<b>CO5</b>	Construct programs using the suitable Collection classes and interfaces for efficient modelling of the objects and entities of the program. (Apply)
<b>CO6</b>	Develop applications that use file input and output. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M		S				M	M			S	S	S
<b>CO2</b>	S	M	M		S				M	M			S	S	S
<b>CO3</b>	S	M	M		S				M	M			S	S	S
<b>CO4</b>	S	M	M		S				M	M			S	S	S
<b>CO5</b>	S	M	M		S				M	M			S	S	S
<b>CO6</b>	S	M	M		S				M	M			S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

Overview of Object-Oriented Programming and its need, Java Programming Elements: Classes and Objects, Data types, Constructors, Input-Output Handling, Control structures, Method overloading and overriding, Abstraction and Inheritance, Interfaces, understanding final and static: classes, blocks and methods, Packages.

Exception Handling: Types of Exceptions, Exception classes, try, catch, throw, throws and finally, Exception Handling with Method Overriding, Custom Exceptions, finalize.

Multithreaded Programming: Introduction to multitasking through processes and threads, creating threads, thread life cycle, thread scheduling, thread priorities, daemon thread, synchronization.

Garbage Collection, Runtime class and Memory management in Java.

String handling: String, String Buffer, StringBuilder and tokenization.

Generics, Lambda expressions.

The Collections framework: List, Set and Map interfaces and the corresponding implementation classes, Enumerator and iterators.

Event handling: Event, Listeners and adapter classes, anonymous inner classes.

Abstract Windowing Toolkit (AWT): Button, Label, Checkbox, Checkbox Group, Text Field, Text Area, Choice, List, Menu, Panel, Scrollbar and Swing components, Layout managers, Complex Components.

MVC and introduction to creational, structural design patterns and behavioural patterns.

File I/O: Streams, Readers and Writers, Pipes and Filters, Random Access, Scanner.

## **Reading List:**

1. Herbert Schildt, Java: The Complete Reference, 11<sup>th</sup> edition, Mc Graw Hill, 2019.
2. Kathy Sierra & Bert Bates, Head First Java, 3<sup>rd</sup> edition, O'Reilly, 2005.
3. Robert C Martin, Clean Code, Pearson, 2012.
4. Timothy Budd, Object Oriented Programming with Java, Pearson Education, 2009.
5. Object Oriented Programming with Java, Debasis Samanta, IIT Kharagpur, accessed through: <https://cse.iitkgp.ac.in/~dsamanta/java/index.htm>, Accessed on: August 2021.
6. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns, Addison-Wesley, 1994.

<b>CS1062</b>	<b>Object Oriented Design and Programming lab</b>	<b>PCC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	---	------------	------------------	-----------------

Pre-Requisites:

- i. Problem Solving through Computer Programming (CS1011)
- ii. Problem Solving through Computer Programming lab (CS1012)

**Course Outcomes:** At the end of the course, the student will be able to

<b>CO1</b>	Analyze the fundamental principles of OOP such as encapsulation, inheritance, polymorphism, and abstraction.
<b>CO2</b>	Analyse the core concepts of multi-threading, including threads, processes, concurrency, and parallelism.
<b>CO3</b>	Develop the ability to create custom exception classes to handle application-specific errors.
<b>CO4</b>	Design and develop real-world applications using Java, adhering to OOP principles.

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	L		S				M			L	L	S	
<b>CO2</b>	S	M	L		S				M			L	L	S	
<b>CO3</b>	S	M	L		S				M			L	L	S	
<b>CO4</b>	S	M	L		S				M			L	L	S	

S: Strong correlation, M: Medium correlation, L: Low correlation

**Tentative List of Experiments:**

1. Develop programs to familiarize with Object Oriented Design concepts.
2. Implement programs to illustrate overloading and overriding.
3. Implement programs to familiarize with the Testing and Debugging facilities.
4. Implement abstract classes and Interfaces.
5. Implement programs using Static classes, blocks and methods.

6. Implement programs to compare shallow and deep copy.
7. Develop programs using Exception Handling.
8. Develop programs using multi-threading and synchronization.
9. Implement programs using Generics.
10. Implement programs using Linked data structures, Heaps, priority queues, binary search trees and the interfaces List, Set and Map.
11. Implement programs using String processing facilities offered by String, String Buffer and StringBuilder classes.
12. Develop Event-driven programs for GUI applications that are interactive, responding to events originating from keyboard and mouse.
13. Implement programs using Streams and File I/O for reading and writing the contents in sequential and random order.
14. Given two words of equal length that are in a dictionary, implement an efficient program to transform one word into another word by changing only one letter at a time.
15. Numbers are randomly generated and passed to a method, implement an efficient program to find and maintain the median value as new values are generated.
16. Given an N x N matrix of positive and negative integers, implement an efficient program to find the sub-matrix with the largest possible sum.
17. Case study/Mini project.

### **Reading List:**

1. Herbert Schildt, Java: The Complete Reference, 11<sup>th</sup> edition, Mc Graw Hill, 2019.
2. Kathy Sierra & Bert Bates, Head First Java, 3<sup>rd</sup> edition, O'Reilly, 2005.
3. Robert C Martin, Clean Code, Pearson, 2012.
4. Timothy Budd, Object Oriented Programming with Java, Pearson Education, 2009.
5. Object Oriented Programming with Java, Debasis Samanta, IIT Kharagpur, accessed through: <https://cse.iitkgp.ac.in/~dsamanta/java/index.htm>, Accessed on: August 2021.
6. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns, Addison-Wesley, 1994.

<b>HS1011</b>	<b>English for Engineers - 1</b>	<b>HSC</b>	<b>2 – 0 – 0</b>	<b>2 Credits</b>
---------------	----------------------------------	------------	------------------	------------------

**Pre-requisites: None**

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	To develop a strong foundation in grammar
<b>CO2</b>	To develop vocabulary and to write effective paragraphs and formal letters
<b>CO3</b>	To improve reading comprehension and team-skills/collaborative skills
<b>CO4</b>	To cultivate interpretive and critical thinking skills

### **Detailed Syllabus:**

Basics of Language: Tense, Concord, Error detection, Reading Comprehension

Writing: Paragraphs, Precis writing, Formal letters, and Email etiquette

Interpretation and Critical Thinking: Cross cultural communication, Identifying biases, Interpretation of visual data and information, and Logical reasoning

Understanding Audience/Profiling Readers, Introduction to workplace communication, Group Contract/Team Contract, Presentation skills, and Techniques to enhance listening skills

### **Reading List:**

1. Anderson, Marilyn, Pramod K. Nayar, and Madhucchanda Sen. Critical Thinking,
2. Academic Writing and Presentation Skills. Pearson Education, 2008.
3. Emden, Joan van. Effective Communication for Science and Technology. Macmillan Education UK, 2001.
4. Murphy, Raymond. Intermediate English Grammar. Cambridge University Press, 2014.
5. Narayanaswami, V. R. Strengthen Your Writing. Orient Longman Private Limited, 2005.
6. Sharma, Sangeetha and Binoth Mishra. Communication Skills for Engineers and Scientists. PHI, 2023
7. Aarts, Bas. Oxford Modern English Grammar. Oxford University Press, 2011.
8. Blake, Gary. The Elements of Technical Writing. Pearson, 2000
9. Carlisle, Joanne and Melinda S. Rice. Improving Reading Comprehension Research-based Principles and Practices. York Press, 2002.
10. Carter, Ronald and Michael McCarthy. Cambridge Grammar of English: A

Comprehensive Guide. Cambridge University Press, 2006.

11. Carter, Ronald, Rebecca Hughes, and Michael McCarthy. Exploring Grammar in Context: Upper-intermediate and Advanced. Cambridge University Press, 2000.
12. Dobelli, Rolf. The Art of Thinking Clearly: Better Thinking, Better Decisions. Sceptre, 2013.
13. Eastwood, John. Oxford Guide to English Grammar. Oxford University Press, 1994. Rolf, Dobelli.
14. <https://learnenglish.britishcouncil.org/>

## II Year Syllabus

<b>MA2031</b>	<b>Probability, Statistics and Stochastic Processes</b>	<b>BSC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---	------------	------------------	------------------

**Pre-requisites:** None.

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Infer the probability of events with an understanding of the axioms and theorems in probability theory. (Apply)
<b>CO2</b>	Apply standard techniques to compute moments of random variables modelling different distributions. (Apply)
<b>CO3</b>	Prove standard results, theorems and inequalities in probability and statistical theories. (Apply)
<b>CO4</b>	Set up and test a hypothesis by identifying and applying a suitable statistical test. (Apply)
<b>CO5</b>	Model a system or a component using a suitable stochastic process and analyze its performance metrics. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M	L									S	M	M
<b>CO2</b>	S	M	M	L									S	M	M
<b>CO3</b>	S	M	M	L									S	M	M
<b>CO4</b>	S	M	M	L									S	M	M
<b>CO5</b>	S	S	M	L									S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Probability Theory:**

Introduction to Probability, definitions, scope and examples of probability; sample spaces and events; axiomatic definition of probability; joint and conditional probabilities; independence, total probability; Bayes' rule and applications. Definition of random variables, continuous and discrete random variables; cumulative distribution function (cdf) for discrete and continuous random variables; probability mass function (pmf); probability

density functions (pdf) and properties; expectation: mean, variance and moments of a random variables, Moment generating and Characteristic functions. Special distributions (Binomial, Hypergeometric, Poisson, Uniform, Exponential, Normal, Chi-square), Markov Inequality, Chebyshev's inequality, Statistic estimation of parameters by maximum Likelihood Estimation method.

### **Hypothesis Testing:**

Testing of Hypothesis, Null and alternative hypothesis, level of significance, one-tailed and two-tailed tests, tests for large samples (tests for single mean, difference of means, single proportion, difference of proportions), tests for small samples (t-test for single mean and difference of means, F-test for comparison of variances), Chi-square test for goodness of fit, analysis of variance (one way classification with the samples of equal and unequal sizes), correlation and regression. Multiple and partial correlation, rank correlation and Karl Pearson coefficient of correlation, regression.

### **Stochastic Processes:**

Definition and classification of general stochastic processes. Markov Chains: definition, transition probability matrices, classification of states, limiting properties, Discrete Time Markov Chains: Chapman-Kolmogorov equations, Ergodicity, Reducibility/Irreducibility, Time reversible chains, Applications of Markov chains in Queuing models, Continuous Time Markov Chains: Birth-Death processes, Poisson processes, M/M/1 Queuing Models.

### **Reading List:**

1. R.A. Johnson, Miller and Freund, Probability and Statistic for Engineers, 9th Edition, Pearson Publishers, 2017.
2. Freund, Modern elementary statistics, PHI, 2006.
3. S.C. Gupta and V.K. Kapoor, Fundamentals of Mathematical Statistics, 2006.
4. K.V. Iyer, Karthick Seshadri, Discrete Mathematics, Cengage, First Edition, 2022.
5. U.N. Bhat and Gregory K. Miller, Elements of Applied Stochastic Processes, John Wiley and Sons, 2002.
6. Sheldon M. Ross, "Introduction to Probability and Statistics for Engineers and Scientists", Academic Press, (2009).

<b>EC1541</b>	<b>Microprocessors</b>	<b>ESC</b>	<b>2 – 0 – 0</b>	<b>2 Credits</b>
---------------	------------------------	------------	------------------	------------------

**Pre-requisites:** Computer Organization and Architecture (CS1021)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Understand the evolution of microprocessors
<b>CO2</b>	Understand assembly language programming basics for 8085
<b>CO3</b>	Understand the basic serial I/O and interrupt mechanism used in 8085
<b>CO4</b>	Understand the data transfer techniques using microprocessor
<b>CO5</b>	Understand the typical 16-bit microprocessor

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	L	L									S	S	S
<b>CO2</b>	S	M	L	L									S	S	S
<b>CO3</b>	S	M	L	L									S	S	S
<b>CO4</b>	S	M	L	L									S	S	S
<b>CO5</b>	S	M	L	L									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

### Detailed Syllabus:

**Introduction to Microprocessors:** History and Evolution, Microcomputer Architecture, Intel 8085 Microprocessor, Register Architecture, Bus Organization, Registers, ALU, Control section, Instruction set of 8085, Instruction format, Addressing modes, Types of Instructions.

**Assembly Language Programming and Timing Diagram:** Assembly language programming in 8085, Macros, Labels and Directives, Microprocessor timings, Micro instructions.

**Serial I/O and Interrupts:** Serial I/O using SID, SOD. Interrupt in 8085, RST instruction, issues in implementing interrupts, Multiple interrupts and priorities, Daisy chaining, interrupt handling in 8085, Enabling, Disabling & masking of interrupts.

**Data Transfer Techniques:** Data transfer techniques, Parallel & Programmed data transfer using 8155. Programmable parallel ports & handshake input/output, Asynchronous and Synchronous data transfer using 8251.

**Architecture of Typical 16-Bit Microprocessors:** Introduction to a 16-bit microprocessor, Memory address space and data organization, Segment registers and Memory segmentation, generating a memory address, I/O address space, addressing modes.

**Reading List:**

1. R.S. Gaonkar, Microprocessor Architecture, Programming & Applications with the 8085/8080A, 5th Ed., Prentice Hall, 2002.
2. A.H. Mukhopadhyay, Microprocessor, Microcomputer and Their Applications, 3<sup>rd</sup> Edition Alpha Science International, Ltd.
3. M. Rafiquzzman: Microprocessors: Theory & Applications (Intel & Motorola), PHI.
4. Berry.B. Bray INTEL 8086/88, 80186, 286, 386, 486, Pentium Pro & Pentium IV.
5. D. V. Hall, Microprocessor and Interfacing', 3rd Edition, TMH.

<b>EC1542</b>	<b>Microprocessors Lab</b>	<b>ESC</b>	<b>0 – 1 – 2</b>	<b>2 Credits</b>
---------------	----------------------------	------------	------------------	------------------

**Pre-requisites:** Computer Organization and Architecture (CS1021)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Write assembly language and C programs for arithmetic operations using 8085
<b>CO2</b>	Interface LED, ADC and DAC modules with microprocessor-based system
<b>CO3</b>	Interface stepper motor, Keyboard and memory

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	L	L									S	S	S
<b>CO2</b>	S	M	L	L									S	S	S
<b>CO3</b>	S	M	L	L									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Tentative List of Experiments:**

1. Write a program using 8085 Microprocessor for Decimal, Hexadecimal addition and subtraction of two Numbers.
2. Write a program using 8085 Microprocessor for addition and subtraction of two BCD numbers.
3. To perform multiplication and division of two 8-bit numbers using 8085.
4. To find the largest and smallest number in an array of data using 8085 instructions set. Interfacing
5. To write a program to arrange an array of data in ascending and descending order.
6. To convert given Hexadecimal number into its equivalent ASCII number and vice versa using 8085 instructions set.
7. To write a program to initiate 8251 and to check the transmission and reception of character.

8. To interface a 7 segment LED with 8085.
9. To interface a stepper motor with 8085.
10. To interface a DAC and ADC with 8085.
11. Case study/Mini project.

**Reading List:**

1. R.S. Gaonkar, Microprocessor Architecture, Programming & Applications with the 8085/8080A, 5th Ed., Prentice Hall, 2002.
2. A.H. Mukhopadhyay, Microprocessor, Microcomputer and Their Applications, 3<sup>rd</sup> Edition Alpha Science International, Ltd.
3. M. Rafiquzzman: Microprocessors: Theory & Applications (Intel & Motorola), PHI.
4. Berry. B. Bray INTEL 8086/88, 80186, 286, 386, 486, Pentium Pro & Pentium IV.
5. D. V. Hall, Microprocessor and Interfacing', 3rd Edition, TMH.

<b>CS2011</b>	<b>Discrete Event Simulation</b>	<b>ESC</b>	<b>1 – 0 – 0</b>	<b>1 Credits</b>
---------------	----------------------------------	------------	------------------	------------------

This course introduces students to the underlying theory, motivation and tools and techniques to construct discrete event simulations to model and analyze the performance of a computing system or its component.

**Pre-requisites:**

- i. Probability, Statistics and Stochastic Processes (MA2031)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Construct discrete event simulation models for characterizing a computing system or its component. (Apply)
<b>CO2</b>	Verify and validate discrete event simulation models (Apply)
<b>CO3</b>	Implement discrete event simulation models using a suitable simulation software and analyze the simulation results using suitable performance metrics. (Analyze)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	L	L									S	S	S
<b>CO2</b>	S	M	L	L									S	S	S
<b>CO3</b>	S	S	M	M	S					S		S	S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

Introduction to Simulation: Scenarios requiring simulations, Merits and demerits of Simulation, Applications, System – environment, components and models, Types of models, Discrete event system simulation, Examples involving queuing and inventory systems.

Simulation Modelling – Types of Simulations: Monte-Carlo, Trace-driven, Discrete Event Simulations: Event Set Algorithms, Random number generation, Random variate generation, Testing random numbers, Analysis of Simulation Results.

Selection of Simulation Software, Examples such as SIMUL8, GPSS, Arena, Network Simulator, Quantum Simulators.

**Reading List:**

1. Jerry Banks, Hon S Carson, Barry L Nelson, David M Nicol, "Discrete Event Simulation 5<sup>th</sup> Edition, Pearson, 2010.
2. Raj Jain, The Art of Computer Systems Performance Analysis, Wiley, 2008.

<b>CS2021</b>	<b>Operating Systems</b>	<b>PCC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Computer Organization and Architecture (CS1021)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Construct process manager with a comprehension of scheduling policies, deadlocks, synchronization and the associated kernel data structures. (Apply)
<b>CO2</b>	Construct memory manager with a comprehension of the issues involved in memory management and management strategies especially for Virtual memory environments. (Apply)
<b>CO3</b>	Design and develop System Call Interface for a given File System. (Apply)
<b>CO4</b>	Design and develop Device Driver for disk access in the xv6. (Apply)
<b>CO5</b>	Analyze the design of Operating System Sub-Components of xv6 (x86 version) with an understanding of the sub-components like Process, Memory, and I/O managers and the interplay among these sub-components. (Analyze)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M		S				M				S	S	M
<b>CO2</b>	S	M	M		S				M				S	S	M
<b>CO3</b>	S	M	M		S				M				S	S	M
<b>CO4</b>	S	M	M		S				M				S	S	M
<b>CO5</b>	S	S	S	M	S				M				S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Processes:**

Introduction to operating systems: Process abstraction, Process states, CPU Data Structures for process control. System calls for process management: fork, exec, exit, wait, open, read, write, close. Process Execution, Execution of function call, kernel/user modes, trap instruction, mechanism of context switch.

Scheduling policies: Criteria Considered for CPU Scheduling, scheduling using FIFO, SJF, STCF, RR, MLFQ methods.

Inter Process Communication: IPC, Shared Memory, Signals, Sockets, Pipes and Message Queues. Introduction to XV6.

Process Control in XV6: Process abstraction, Process state transition example. Process management system calls: Fork, Exec, Exit, Wait. Trap handling in XV6, Scheduling and Context switching in XV6, User process creation in XV6.

### **Memory Management:**

Introduction to virtual memory: Goals of memory virtualization, Memory allocation system calls. Mechanism of address translation: Role of OS in translation, Segmentation. Paging: Page table, Multiple page tables. Demand paging: Page faults, Page replacement policies. Memory allocation and free space management algorithms: Variable size allocations and Fixed size allocations.

Virtual Memory and Paging in XV6, Memory management and User processes in XV6.

### **Concurrency:**

Introduction to threads and concurrency: Single threaded process, multiple threaded process, scheduling threads, Race condition and Synchronization. Locks: Building a lock, different types of locking mechanisms. Condition Variables: producer-consumer problem. Semaphores. Concurrency bugs: non-deadlock bugs, deadlock bugs, conditions for deadlock, prevention mechanisms, deadlock avoidance, deadlock detection, recovery from deadlock. Locking in XV6: spin locks, disabling interrupts. Sleep/wakeup functionality in XV6.

### **I/O and file systems:**

**Communication with I/O devices:** Introduction to input/output devices, Interrupts, Direct Memory Access, Device driver. Files and Directories: File abstraction, Directory tree, Hard links and soft links, mounting a file system, Memory mapping a file. File system implementations: File system, I-node table, File Allocation Table (FAT), Virtual file system. Hard disk internals: Disk scheduling, SSTF, SCAN Algorithm, SPTF.

Device drivers and block I/O in XV6: File system and I/O in XV6, Disk blocks and buffers, Logging layer. File system in XV6: Disk layout, in memory data structure, I-node function, creating a file, system calls: open, link, file read.

**Security in Operating System:** Goals of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Revocation of Access Rights, Language-Based Protection, Capability-Based Systems, The Security Problem, User Authentication, Program Threats, System Threats, Securing Systems and Facilities, Intrusion Detection, Cryptography, Computer-Security Classifications.

### **Reading List:**

1. Remzi H. Arpaci-Dusseau Andrea C. Arpaci-Dusseau, "Operating Systems: Three Easy Pieces", Amazon Digital Services, 2015.

2. xv6 a simple, Unix-like teaching operating system, Russ Cox Frans Kaashoek Robert Morris, Available From: <https://pdos.csail.mit.edu/6.828/2014/xv6/book-rev8.pdf>, 2022
3. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, 10th edition, Wiley, 2008.

<b>CS2022</b>	<b>Operating Systems Lab</b>	<b>PCC</b>	<b>0 – 1 – 2</b>	<b>2 Credits</b>
---------------	------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Computer Organization and Architecture (CS1021)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Implement elementary UNIX system commands. (Apply)
<b>CO2</b>	Develop shell scripts for data wrangling (Apply)
<b>CO3</b>	Design and develop user level thread library. (Apply)
<b>CO4</b>	Design and implement a file system. (Apply)
<b>CO5</b>	Design and implement a memory manager module. (Apply)
<b>CO6</b>	Design and implement a process manager module. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M		S				M	M			S	S	S
<b>CO2</b>	S	M	M		S				M	M			S	S	S
<b>CO3</b>	S	M	M		S				M	M			S	S	S
<b>CO4</b>	S	M	M		S				M	M			S	S	S
<b>CO5</b>	S	M	M		S				M	M			S	S	S
<b>CO6</b>	S	M	M		S				M	M			S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Tentative List of Experiments:**

1. Write Command Interpreter Programs which accepts some basic Unix commands and displays the appropriate result. Each student should write programs for at least six commands.
2. Study the concept of Signals and write a program for Context Switching between two processes using alarm signals.
3. Write a program to do memory management (allocation/freeing) for fixed chunk sizes using a larger chunk of memory allocated using mmap builtin function. Extend it to manage dynamic memory allocation as done with heap memory.

4. Study pthreads and implement the following: Write a program which shows the performance improvement in using threads as compared with process. (Examples like Matrix Multiplication, Hyper quicksort, Merge sort, Traveling Salesperson problem)
5. Create your own thread library, which has the features of pthread library by using appropriate system calls (UContext related calls). Containing functionality for creation, termination of threads with simple round robin scheduling algorithm and synchronization features.
6. Implement all CPU Scheduling Algorithms using your thread library
7. Study the concept of Synchronization and implement the classical synchronization problems using Semaphores, Message queues and shared memory (minimum of 3 problems)
8. A complete file system implementation inside a disk image file.  
Unix Scripting:
9. Create a bash script to read 2 positive numbers and print out all the numbers between them. You need to handle the following errors
  - a. Both numbers must be positive
  - b. The numbers must be unique
  - c. The first number must be less than second
  - d. The loop runs from first number to the second number
10. Write a shell script to modify the string “Strawberry” from the lines starting with “Apple” in all the sub directories of a given folder with a given string “Banana”
11. Write a shell script to add specific hostname ip address combination in all the computers given in command line in the /etc/hosts file
12. Write a script for sorting a huge file by distributing it on to multiple machines (using scp and ssh) and collecting the data in a single file at the host. Report the times of sort utility vs your script for different sizes of the input file.
13. Write a script to take two files and merge the two files based on common column (similar to database join). Use the first line in each of the files to be the column header of the data.
14. Case study/Mini project.

### Reading List:

1. Richard Stevens, Stephen Rago, Advanced Programming in the UNIX Environment, Third edition, Addison Wesley, 2013.
2. Remzi H. Arpaci-Dusseau Andrea C. Arpaci-Dusseau, Arpaci-Dusseau Books, Operating Systems: Three Easy Pieces, Amazon Digital Services, 2015.
3. xv6 a simple, Unix-like teaching operating system, Russ Cox Frans Kaashoek Robert Morris, Available From: <https://pdos.csail.mit.edu/6.828/2014/xv6/book-rev8.pdf>, Accessed on: August 2021.
4. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, 10th edition, Wiley, 2008.

<b>CS2031</b>	<b>Design and Analysis of Algorithms</b>	<b>PCC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--	------------	------------------	------------------

**Pre-requisites:**

- i. Data Structures and Algorithms (CS1051)
- ii. Data Structures and Algorithms lab (CS1052)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Infer which algorithm design paradigm is suitable for solving a given problem. (Analyze)
<b>CO2</b>	Construct algorithms for a given complex engineering problem using an appropriate design paradigm such as divide and conquer, dynamic programming, greedy, backtracking and branch-and-bound. (Apply)
<b>CO3</b>	Construct correctness proofs for algorithms. (Apply)
<b>CO4</b>	Infer the asymptotic time and space complexities of algorithms. (Apply)
<b>CO5</b>	Assess the impact of the choice of data structure on the performance of algorithms. (Apply)
<b>CO6</b>	Construct algorithms for graph-based problems by modeling entities as graphs and reducing the solution to standard graph theoretic algorithms. (Apply)
<b>CO7</b>	Construct reductions between problems with a comprehension of the complexity classes namely P, NP, NP-Complete & NP-Hard and their relationships. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	S	M	L									S	S	S
<b>CO2</b>	S	M	L	L									S	S	S
<b>CO3</b>	S	M	L	L									S	S	S
<b>CO4</b>	S	M	L	L									S	S	S
<b>CO5</b>	S	M	L	L									S	S	S
<b>CO6</b>	S	M	L	L									S	S	S
<b>CO7</b>	S	M	L	L									S	S	S

Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

Introduction to Algorithm Design and Analysis, Asymptotics, Correctness Proofs for Algorithms.

Divide and Conquer Algorithms: Solving Recurrences, Recurrence Trees, Substitution method, Master's Theorem, Merge Sort, Quick Sort and Randomized Quicksort with their analyses, Maximum sub-array problem, Linear time Selection, Strassen's Matrix Multiplication, Multiplication of Large integers.

Greedy Algorithms: Design scenarios calling for a greedy design, Making Change problem, Activity Selection Problem, Fractional Knapsack problem, Prim's and Kruskal's algorithms for Minimum Spanning Tree construction, Dijkstra's algorithm for finding single source shortest paths.

Dynamic Programming: Design scenarios calling for a dynamic programming design, Chained Matrix Multiplication problem, 0/1 Knapsack problem, Travelling Salesperson Problem, Optimal Binary Search Tree, Floyd Warshall's Algorithm, Vertex Cover of a Tree.

Backtracking: N-Queen's problem, Enumeration of Independent Sets of a problem, Graph Coloring problem, Robotic path finding problem in an unknown terrain.

Branch and Bound: TSP. Set Cover and 0/1 Knapsack problem.

String algorithms: KMP Algorithm, Boyer Moore Algorithm and Rabin-Karp Algorithm.

Flow-based Algorithms: Max-flow Min-cut theorem, augmenting paths and finding min-cuts.

Complexity classes: P, NP, NP-Complete and NP-Hard, Standard reductions.

## **Reading List:**

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.
2. Michael T. Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis and Internet Examples, Second Edition, Wiley-India, 2006.
3. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, Second Edition, Universities Press, 2011.
4. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, First Edition, PHI, 2000.
5. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, Algorithms, First Edition, McGraw Hill Education, 2006.
6. Sara Baase, Computer algorithms: Introduction to Design and Analysis, Second Edition, Addison Wesley publication, 1998.

<b>CS2032</b>	<b>Algorithms lab</b>	<b>PCC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	-----------------------	------------	------------------	-----------------

**Pre-requisites:**

- i. Data Structures and Algorithms (CS1051)
- ii. Data Structures and Algorithms lab (CS1052)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Implement, debug, test and run algorithms by leveraging suitable flow of control constructs and syntactic units of the programming language. (Analyze)
<b>CO2</b>	Construct efficient algorithms for problems by orchestrating sub-algorithms for solving problems such as sorting, searching, selection and / or arithmetic computations. (Analyze)
<b>CO3</b>	Infer which design paradigm and data structure is an optimal choice to construct an algorithm to solve a given complex engineering problem. (Analyze)
<b>CO4</b>	Construct efficient algorithms and their implementations for solving complex engineering problems by demonstrating problem-solving skills and out-of-the-box algorithmic thinking. (Analyze)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M		S				M			L	S	M	M
<b>CO2</b>	S	M	M		S				M			L	S	M	M
<b>CO3</b>	S	M	M		S				M			L	S	M	M
<b>CO4</b>	S	M	M		S				M			L	S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Tentative list of Laboratory Exercises:**

Students will be required to implement the algorithms taught in the class and will also be required to design, analyze and implement algorithms for unknown problems. Further, a student will be required to implement a mini project for a complex engineering problem which will involve multiple design-options, trade-offs, algorithm and/or data structure design and analysis.

1. Problems based on Searching.
2. Problems based on Sorting.
3. Problems based on Selection.

4. Problems requiring solutions based on divide and conquer algorithms.
5. Problems that can be solved using greedy algorithms.
6. Problems requiring a dynamic programming-based solution.
7. Problems requiring backtracking.
8. Optimization problems admitting branch and bound based solutions.
9. Flow based algorithms.
10. Pattern matching algorithms.
11. Problems requiring graph-based solutions.
12. Mini project to unify the concepts learnt in the course.
13. Case study/Mini project.

### **Reading List:**

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.
2. Michael T. Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis and Internet Examples, Second Edition, Wiley-India, 2006.
3. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, Second Edition, Universities Press, 2011.
4. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, First Edition, PHI, 2000.
5. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, Algorithms, First Edition, McGraw Hill Education, 2006.
6. Sara Baase, Computer algorithms: Introduction to Design and Analysis, Second Edition, Addison Wesley publication, 1998.

<b>CS2041</b>	<b>Theory of Computation</b>	<b>PCC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	------------------------------	------------	------------------	------------------

**Pre-requisites:** Discrete Mathematics (MA1051)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Construct, compare and analyze computational machine models for solving problems and recognizing languages. (Analyze)
<b>CO2</b>	Infer the properties of languages, automata and grammars with the help of rigorous proofs. (Analyze)
<b>CO3</b>	Infer the limitations of the computational models and prove the existence of such limitations. (Analyze)
<b>CO4</b>	Construct algorithms for solving problems and show the correctness of algorithms using different computational machine models. (Analyze)
<b>CO5</b>	Identify the dependency of application domains like Compiler Design on the computational models. (Analyze)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	S	M										S	M	M
<b>CO2</b>	S	S	M										S	M	M
<b>CO3</b>	S	S	M										S	M	M
<b>CO4</b>	S	S	M										S	M	M
<b>CO5</b>	S	S	M										S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Finite State Automata (FSA):** Alphabets, Strings, Languages, Deterministic FSA (DFSA), non-deterministic FSA (NFSA) with Epsilon-Transitions, Uses of  $\epsilon$  Transitions. Regular expressions, Finite Automata and Regular Expressions, Conversion among Regular Expressions, DFSA and NFSA, Applications of Regular Expressions, Applications of Regular Expressions in UNIX and Lexical Analysis, Algebraic Laws for Regular Expressions, Laws Involving Closures, Subset Construction Algorithm, FSA Minimization, Regular Languages and properties, Pumping Lemma, Myhill-Nerode Theorem.

**Context Free Grammars** - Derivations Using a Grammar, Leftmost and Rightmost Derivations, The Language of a Grammar, Sentential Forms, Parse Trees, eliminating useless symbols, Applications of Context-Free Grammars, Parsers, The YACC Parser Generator, Ambiguity in Grammars and Languages, Ambiguous Grammars, Removing Ambiguity from Grammars. Normal Forms: Chomsky Normal Form and Greibach Normal Form, Chomsky Hierarchy, Context Free Languages (CFLs), properties, Pumping Lemma for CFLs, CYK Algorithm.

**Push Down Automata (PDA):** Definition, Instantaneous Descriptions of a PDA, The Languages of a PDA, Acceptance by Final State, Acceptance by Empty Stack, Equivalence of PDAs and CFGs, Deterministic Context Free Languages.

**Turing Machines (TMs):** Definition, Instantaneous Descriptions for the Turing Machines, Transition Diagrams for Turing Machines, The Language of a Turing Machine, Turing Machines and Halting, Configurations: TM as a computing device and as a recognition device. Programming Techniques for Turing Machines: Storage in the State, Multiple Tracks, Shifting Over, Subroutines, Extensions to the Basic Turing-Machines, Multiple Turing Machines, Computable Functions. Multi-Tape Turing Machines, Linear Bounded Automata.

Recursive and recursively enumerable languages. Undecidability of Halting Problem. Reductions, Theory of NP Completeness.

#### **Reading List:**

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D Ullman, Introduction to Automata Theory, Languages and Computation, Second Edition, Pearson, 2001.
2. Michael Sipser, Introduction to Theory of Computation, Third Edition, Course Technology, 2012.
3. Automata and Computability, Dexter C. Kozen, First Edition, Springer Publishers, 2007.
4. Elements of the Theory of Computation, H. R. Lewis and C.H. Papadimitriou, First Edition, Prentice Hall Publishers, 1981.
5. J. Martin, Introduction to Languages and the Theory of computation, Third Edition, Tata Mc Graw Hill, 2007.
6. Peter Linz, An Introduction to Formal Languages and Automata, Sixth Edition, Jones & Bartlett Learning, 2017.

<b>CS2051</b>	<b>Data Base Management Systems</b>	<b>PCC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-------------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Data Structures and Algorithms (CS1051)
- ii. Data Structures and Algorithms Lab (CS1052)
- iii. Operating Systems (CS2021)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Construct database schema using Database models at conceptual level identifying entities and relationships among entities using E-R and relational models. (Apply)
<b>CO2</b>	Construct database and implement queries using SQL constructs for a given requirement specification. (Apply)
<b>CO3</b>	Construct database design using Normalization and Functional Dependencies to store information without redundancy. (Analyze)
<b>CO4</b>	Design Data Storage Structures and Indexing at physical level to build fast and reliable storage systems and enable quick access to data. (Analyze)
<b>CO5</b>	Apply concurrency control mechanisms with the help of locking, stamping and optimistic methods, and recovery mechanisms to maintain database integrity. (Apply)
<b>CO6</b>	Apply strategies for transaction management with a comprehension of the ACID properties to be maintained. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	L	L	S				M	L			S	S	L
<b>CO2</b>	S	M	L	L	S				M	L			S	S	L
<b>CO3</b>	S	M	S	L	S				M	L			S	S	M
<b>CO4</b>	S	M	S	L	S				M	L			S	S	M
<b>CO5</b>	S	M	S	L	S				M	L			S	S	M
<b>CO6</b>	S	M	S	L	S				M	L			S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

## Detailed Syllabus:

**Introduction:** Basic concepts of database systems, Database System Applications, Purpose of Database Systems, Data View, Database Languages, Characteristics of a database approach, Advantages of DBMS over file system, Database Design, Database Engine, Database and Application Architecture, Database Users and Administrators, History of Database systems

**Database models:** ER-model: Features of the E-R model, Database design using the E-R model, Relational model: Relational Database Schemas, Keys, Integrity constraints over relations, E-R model to relational schema, relational algebra, relational calculus.

**SQL:** Form of a basic SQL Query, Insert, Delete and Update database, Set Operations, Nested queries, Aggregate Functions, Null Values, Integrity Constraints, Join Expressions – Natural join, Equi-join, Inner join, Outer join, Views, Triggers.

**Database design:** Functional-Dependency, Soundness and completeness of Armstrong's axioms, Normal Forms, Loss-less decomposition, Dependency preservation, Decomposition Using Functional Dependencies, Multivalued Dependencies and 4NF, Join dependencies and 5NF, Decomposition Algorithms.

**Data Storage Structure and Indexing:** Memory hierarchy, Redundant Arrays of Independent Disks, File organization and indexing, Index Data Structures: Hash-based indexing and Tree-based Indexing (B+ trees and B-trees), Introduction to Query optimization and execution.

**Transactions, concurrency control and recovery:** ACID properties, Transactions and Schedules, Concurrent execution of transactions, Lock-Based Protocols, Deadlock Handling, Concurrency control without locking, Introduction to crash recovery, ARIES, Write-Ahead log protocol, checkpointing, recovery from a system crash, security, authorization and access control.

## Reading List:

1. A Silberschatz, H Korth and S Sudarshan, Database System Concepts, 7th edition, McGraw-Hill, 2010.
2. R Elmasri, S Navathe, Fundamentals of Database Systems, 7th edition, Pearson, 2016.
3. R Ramakrishnan, J Gehrke, Database Management Systems, 3rd edition, McGraw-Hill, 2003.
4. C J Date, An Introduction to Database Systems, Pearson/Addison Wesley, 2003.

<b>CS2052</b>	<b>Data Base Management Systems lab</b>	<b>PCC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	---	------------	------------------	-----------------

**Pre-requisites:**

- i. Data Structures and Algorithms (CS1051)
- ii. Data Structures and Algorithms Lab (CS1052)
- iii. Operating Systems (CS2021)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Design database schema and Implement SQL queries for a given requirement specification. (Apply)
<b>CO2</b>	Assess the impact of different query execution plans and access paths on query performance. (Analyze)
<b>CO3</b>	Implement database maintenance and control using authorization access control, transaction and concurrency management and recover constructs. (Apply)
<b>CO4</b>	Design and develop an application using stored procedures for a given requirement specification. (Apply)
<b>CO5</b>	Design and develop an application with database connectivity for a given requirement specification. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M	L	S			S	S	M			S	S	L
<b>CO2</b>	S	S	S	L	S			S	S	M			S	S	L
<b>CO3</b>	S	M	M	L	S			S	S	M			S	S	L
<b>CO4</b>	S	M	M	L	S			S	S	M			S	S	L
<b>CO5</b>	S	M	M	L	S	S	S	S	S	M	L	L	S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Tentative List of Experiments:**

1. Familiarization with installation of any database management system.
2. Design a database schema and create a database through implementation of Data Definition Language commands - create, alter, drop, rename and truncate.

3. Querying and modifying the database using Data Manipulation Language commands -select, insert, update, delete.
4. Implementation of Aggregate Functions – sum, avg, min, max, count. Use group-by and having clause.
5. Perform join operations – natural join, equi-join, outer join, left outer join, right outer join, inner join and assess the impact of query plans on the performance of join heavy queries.
6. Perform set operations - union, intersection, set difference.
7. Implementation of corelated sub-queries and nested queries.
8. Creating and querying views and materialized views.
9. Implementation of Data Control Language commands – grant and revoke.
10. Implementation of Transaction Control Language commands - commit, save point, and rollback.
11. Implementation of PL/SQL block using variables, operators, data types, control structures.
12. Implementation of PL/SQL stored procedures.
13. Implementation of PL/SQL stored functions.
14. Implementation of PL/SQL cursors structures.
15. Implementation of PL/SQL exception handlers.
16. Implementation of triggers.
17. Implementation of an application with database connectivity for storing and retrieving information.
18. Case study/Mini project.

#### **Reading List:**

1. A Silberschatz, H Korth and S Sudarshan, Database System Concepts, 7th edition, McGraw-Hill, 2010.
2. R Elmasri, S Navathe, Fundamentals of Database Systems, 7th edition, Pearson, 2016.
3. R Ramakrishnan, J Gehrke, Database Management Systems, 3rd edition, McGraw-Hill, 2003.
4. C J Date, An Introduction to Database Systems, 3rd edition, Pearson/Addison Wesley, 2003.
5. PLSQL accessed through:  
<https://www.oracle.com/in/database/technologies/appdev/plsql.html>, Accessed on August 2021.
6. MySQL, accessed through: <https://www.mysql.com/>, Accessed on August 2021.
7. PostgreSQL, accessed through: <https://www.postgresql.org/>, Accessed on August 2021.

8. JDBC, accessed through: <https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html>, Accessed on August 2021.

<b>CS2061</b>	<b>Software Development</b>	<b>PCC</b>	<b>2 – 1 – 0</b>	<b>3 Credits</b>
---------------	-----------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Data Base Management Systems (CS2051)
- ii. Data Base Management Systems Lab (CS2052)
- iii. Object Oriented Design and Programming (CS1061)
- iv. Object Oriented Design and Programming Lab (CS1062)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Analyze the basic web components and development environment.
<b>CO2</b>	Analyze a standard way to organize the data using XML/JSON.
<b>CO3</b>	Construct a simple interactive web application using both client-side and server-side scripting
<b>CO4</b>	Build a simple web application using RESTful web service.
<b>CO5</b>	Examine the need of security in developing web applications.

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M	L	M								M	L	L
<b>CO2</b>	S	M	M	L	M				M				S	M	L
<b>CO3</b>	S	M	S	L	S				S		M		S	M	L
<b>CO4</b>	S	M	S	M	S				S		M		S	M	L
<b>CO5</b>	S	M	S	M	S				S		M		S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

Deconstructing the web from URL to website - Different types of web applications, HTML5 – Static vs Dynamic sites, HTML5: headers, Linking, Images, Image map, meta elements, frameset, HTML forms, cascading style sheet, DHTML: object model, Event model, two tier and Three tier architecture, tools to analyze HTTP requests.

UI design for web applications using CSS3.

**Developer Tools:** Introduction - Git, IDE/ Editors, Deployment, Console

**Frameworks:** Mean / Mern / Django

**XML Basics-** Structure- Elements and attributes- Namespaces- Working with DTD Schema- Grouping elements- writing and Parsing XML Document. JSON - Introduction.

**Client-side scripting** – Javascript (ES6+) / JQuery, DOM manipulation through XQuery, Responsive Web Design, Front end framework- Angular JS/ React JS.

**Different kinds of servers** - web servers – Apache & nginx, File servers – time servers- DB servers. Introduction to NodeJS – asynchronous nature of nodejs – simple apps Integrating application with DB – DB drivers- Integrate NodeJS with NOSQL, Integrate NodeJS with SQL.

**Web Architecture & Web services:** MVC introduction- thin clients Vs Thick clients. Web services – Introduction- SOAP, REST – writing a RESTful service (nodejs + express). SOAP Vs REST.

**Web security model** - HTTPS: Goals and Pitfalls - Injection flaws: Cross site scripting, SQL Injection, Command Injection, Cross site request forgery and HTTP header injection - Cookie Flaws and Server Misconfiguration - Session management and user authentication.

### **Tentative List of Experiments:**

Choose a real-life application and incrementally build the full stack application as a case study.

1. Construct a static website using basic HTML5 and CSS3
2. Design XML document for formatting, styling, and schema
3. Construct a RESTful service by utilizing Back-end application
4. Develop a full stack application and deploy using Github/Heroku
5. Simulate SQL injection

### **Reading List:**

1. Jeffrey C. Jackson, Web Technologies – A Computer Science Perspective, Fourth Edition, Pearson Education, 2012.
2. Paul Deitel, Harvey Deitel, Abbey Deitel, Internet and World Wide Web How to Program, Fifth Edition, Pearson, 2011.
3. Leonard Richardson & Sam Ruby, RESTful Web Services, O'Reilly, 2007
4. Anthony, Accomazzo, Murray Nathaniel, Lerner Ari, Fullstack React: The Complete Guide to React JS and Friends, Fullstack.io, 2017.
5. Brown, Ethan, Web Development with Node and Express: Leveraging the JavaScript Stack, O'Reilly Media, 2019.
6. Dayley B., Node.js, MongoDB, and AngularJS Web Development, Addison-Wesley Professional, 2014.
7. Laura Lernay, Rafe Colburn, Jennifer Kyrnir, Mastering HTML, CSS & Javascript Web, BPB Publications, 2016.

<b>CS2062</b>	<b>Software Development lab</b>	<b>PCC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	---------------------------------	------------	------------------	-----------------

**Pre-requisites:**

- i. Data Base Management Systems (CS2051)
- ii. Data Base Management Systems Lab (CS2052)
- iii. Object Oriented Design and Programming (CS1061)
- iv. Object Oriented Design and Programming Lab (CS1062)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Analyze and build dynamic and interactive web application.
<b>CO2</b>	Design and implement database schemas for web applications.
<b>CO3</b>	Develop a RESTful Web Service.
<b>CO4</b>	Apply different security mechanisms to build a robust web application.

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M	L	M				M		M		M	L	L
<b>CO2</b>	S	M	M	L	M				M		M		S	M	L
<b>CO3</b>	S	M	S	L	S				S		M		S	M	L
<b>CO4</b>	S	M	S	M	S				S		M		S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Tentative List of Experiments:**

Choose a real-life application and incrementally build the full stack application as a case study.

1. Construct a static web page structure and appearance with HTML and CSS.
2. Construct a client-side interactive and behavior application with JavaScript.
3. Design XML documents for formatting, styling, and schema.

4. Construct a simple front-end application with asynchronous JavaScript using Web Services (APIs).
5. Construct a web service with server-side JavaScript via Node.js.
6. Storing and retrieving information in a database with SQL/NOSQL databases.
7. Develop a full-stack web application and deploy using GitHub/Heroku.
8. Simulate SQL and other Injection attacks.
9. Case study/Mini project

**Reading List:**

1. Robert W. Sebesta, Programming with World Wide Web, Fourth Edition, Pearson, 2008.
2. David William Barron, The World of Scripting Languages, Wiley Publications, 2000.
3. Dayley B., Node.js, MongoDB, and AngularJS Web Development, Addison-Wesley Professional, 2014.
4. Vainikka J., Full-Stack Web Development using Django REST Framework and React, 2018.

<b>CS2071</b>	<b>Compiler Design</b>	<b>PCC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Data Structures and Algorithms (CS1051)
- ii. Data Structures and Algorithms Lab (CS1052)
- iii. Theory of Computation (CS2041)
- iv. Computer Organization and Architecture (CS1021)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Build a lexical analyser for the given language. (Apply)
<b>CO2</b>	Construct suitable parsers for a given language construct with a comprehension of different parsers and their applicability to different constructs. (Apply)
<b>CO3</b>	Construct syntax directed translation schemes subsequent to the identification of appropriate synthesized and inherited attributes. (Apply)
<b>CO4</b>	Construct intermediate code generators to generate intermediate code in the form of three address code representations. (Apply)
<b>CO5</b>	Construct a code generator and code optimizer with a comprehension of the runtime environment. (Apply)
<b>CO6</b>	Construct system software like assemblers, linkers and loaders with a comprehension of the underlying runtime architecture. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	S	M										S		
<b>CO2</b>	S	M	L										S		
<b>CO3</b>	S	M	L										S		
<b>CO4</b>	S	M	L										S		
<b>CO5</b>	S	S	M										S		
<b>CO6</b>	S	M	L										S		

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

Introduction to System Software: Macro processors, One pass and two pass assemblers, Linkers, Loaders, Types of Loaders, Interpreters, Debuggers, Compilers.

Phases of Compilers - Compiler Construction Tools - Bootstrapping

Lexical analyzer - The Role of the Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, A Language for Specifying Lexical Analyzers.

Parsing - The Role of the Parser, Context-Free Grammars, Top-Down Parsing, Bottom-Up Parsing, Operator-Precedence Parsing, LR Parsers, Using Ambiguous Grammars, Parser Generators.

Syntax-Directed Translation- Syntax-Directed Definitions, Construction of Syntax Trees, Bottom-Up Evaluation of S-Attributed Definitions, L-Attributed Definitions, Top-Down Translation, Bottom-Up Evaluation of Inherited Attributes, Recursive Evaluators, Space for Attribute Values at Compile Time, Assigning Spaces at Compiler-Construction Time, Analysis of Syntax-Directed Definitions.

Type Checking- Type Systems, Specification of a Simple Type Checker, Equivalence of Type Expressions, Type Conversions, Overloading of Functions and Operators, Polymorphic Functions, An algorithm for Unification.

Run-Time Environments - Source Language Issues, Storage Organization, Storage-Allocation Strategies, Access to Nonlocal Names, Parameter Passing, Symbol Tables, Language Facilities for Dynamic Storage Allocation, Dynamic Storage Allocation Techniques, Storage Allocation in Fortran.

Intermediate Code Generation - Intermediate Languages, Declarations, Assignment Statements, Boolean Expressions, Case Statements, Backpatching, Procedure Calls.

Code Generation - Issues in the Design of a Code Generator, The Target Machine, Run-Time Storage Management, Basic Blocks and Flow Graphs, Next-Use Information, A Simple Code Generator, Register Allocation and Assignment, The Dag Representation of Basic Blocks, Peephole Optimization, Generating Code from DAGs, Dynamic Programming Code-Generation Algorithm, Code-Generator Generators.

**Reading List:**

1. Alfred V. Aho, Monical S.Lam, Ravi Sethi, and Jeffrey D. Ullman Compilers - Principles, Techniques and Tools, 2nd Edition, Pearson, 2007.
2. Randy Allen, Ken Kennedy, Optimizing Compilers for Modern Architectures, Morgan Kauffmann, 2001.

<b>CS2072</b>	<b>Compiler Design Lab</b>	<b>PCC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	----------------------------	------------	------------------	-----------------

**Pre-requisites:**

- i. Data Structures and Algorithms (CS1051)
- ii. Data Structures and Algorithms Lab (CS1052)
- iii. Theory of Computation (CS2041)
- iv. Computer Organization and Architecture (CS1021)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Implement a lexical analyzer for the given language using lexical analyzer generators. (Apply)
<b>CO2</b>	Implement suitable parsers for a given language construct with a comprehension of different parsers and their applicability to different constructs. (Apply)
<b>CO3</b>	Implement syntax directed translation schemes subsequent to the identification of appropriate synthesized and inherited attributes. (Apply)
<b>CO4</b>	Implement intermediate code generators to generate intermediate code in the form of three address code representations. (Apply)
<b>CO5</b>	Implement code generators with a comprehension of the runtime environment. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M		S				M			L	S	M	M
<b>CO2</b>	S	M	M		S				M			L	S	M	M
<b>CO3</b>	S	M	M		S				M			L	S	M	M
<b>CO4</b>	S	M	M		S				M			L	S	M	M
<b>CO5</b>	S	M	M		S				M			L	S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

### **Tentative List of Laboratory Exercises:**

1. Implementation of Lexical Analyzer using Lexical analyzer generators like Lex.
2. Removing direct and indirect Left Recursion from grammars.
3. Implementation of top-down parsers.
4. Implementation of Syntax Directed Translators and Translation Schemes.
5. Implementation of SLR parser.
6. Implementation of Canonical LR and LALR parsers.
7. Using parser generators such as YACC to generate parsers.
8. Generation of intermediate code for different syntactic constructs in a high-level language such as expressions, statements, conditionals, loops, procedures, and switch-case
9. Implement the Aho-Johnson's algorithm for code generation for a target run-time.
10. Implement the Sethi-Ullman algorithm for code generation for a target run-time, such that the code uses as few registers as possible.
11. Case study/Mini project

### **Reading List:**

1. Alfred V. Aho, Monical S.Lam, Ravi Sethi, and Jeffrey D. Ullman Compilers - Principles, Techniques and Tools, 2nd Edition, Pearson, 2007.
2. Randy Allen, Ken Kennedy, Optimizing Compilers for Modern Architectures, Morgan Kauffmann, 2001.

<b>CS2081</b>	<b>Artificial Intelligence</b>	<b>PCC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Discrete Mathematics (MA1051)
- ii. Probability, Statistics and Stochastic Processes (MA2031)
- iii. Design and Analysis of Algorithms (CS2031)

**Course Outcomes:** At the end of the course, the students will be able to:

<b>CO1</b>	Solve single agent and multi agent problems by applying the appropriate search techniques. (apply)
<b>CO2</b>	Create logical agents to do inference using first-order logic. (apply)
<b>CO3</b>	Construct an agent that can reason under uncertainty and performs inference in a Bayesian Network. (apply)
<b>CO4</b>	Build a ML model from the data and evaluate its performance. (apply)
<b>CO5</b>	Construct an agent that can learn with complete data and learn with hidden variables. (apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	M	S	S	L	M								S	L	M
<b>CO2</b>	S	S	S	M	M								S	L	L
<b>CO3</b>	S	S	S	M	M								S	L	L
<b>CO4</b>	S	S	S	M	M								S	L	M
<b>CO5</b>	S	S	S	M	M								S	L	L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

Introduction and Scope, Agents and Environments, Concept of Rationality.

Solving problems by Search: Uninformed Search Strategies, Breadth-first search, Depth-first search, Searching with Partial Information, Informed (Heuristic) Search Strategies, Greedy best-first search.

A\* Search: Minimizing the total estimated solution cost, Heuristic Functions, Local Search Algorithms, and Optimization Problems, Online Search Agents, and Unknown Environments.

Adversarial Search: Games, the minimax algorithm, Optimal decisions in multiplayer games, Alpha-Beta Pruning, Evaluation functions, cutting off search, Games that Include an Element of Chance.

Constraint Satisfaction Problem: Definition, Constraint Propagation, Search for CSPs.

Inference In First Order Logic: Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution. The Wumpus World.

Planning: Classical Planning, algorithms for planning as state space search, planning graphs.

Uncertainty: Inference Using Full Joint Distributions, Independence, Bayes Rule, and its Use, The Wumpus World Revisited.

Probabilistic Reasoning: Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distribution, Exact Inference in Bayesian Networks, Approximate Inference in Bayesian Networks.

Introduction to Reinforcement Learning.

### **Reading List:**

1. Stuart Russell, Peter Norvig, "Artificial Intelligence - A Modern Approach", 4/e, Pearson, 2022.
2. Kevin Knight, Elaine Rich and Shivashankar B. Nair, "Artificial Intelligence", Third Edition, McGraw Hill Education, 2017.
3. Deepak Khemani, "First Course in Artificial Intelligence", McGraw Hill Education, 2017

### III Year Core Courses

<b>CS3011</b>	<b>Machine Learning</b>	<b>PCC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-------------------------	------------	------------------	------------------

**Pre-requisites: -**

- i. Modelling and Optimization Techniques (EC2511)
- ii. Probability, Statistics and Stochastic Processes (MA2031)
- iii. Differential and Integral Calculus (MA1011)
- iv. Matrices and Differential Equations (MA1021)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Design learning objectives for classification and regression problems
<b>CO2</b>	Design feature selection strategies using PCA/ SVD/ Spectral Clustering/EM
<b>CO3</b>	Design Sequence Learning Models using generative/discriminative approaches
<b>CO4</b>	Design neural network to solve classification and function approximation problems
<b>CO5</b>	Design efficient deep neural networks

#### Course Articulation Matrix

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	M	S	S	M	M	-	S	-	-	S	-	-	M	S	M
<b>CO2</b>	M	S	M	M	M	-	S	-	-	S	-	-	M	M	M
<b>CO3</b>	M	S	S	M	M	-	S	-	-	S	-	-	M	M	M
<b>CO4</b>	M	M	M	M	M	S	M	-	-	S	S	-	M	M	L
<b>CO5</b>	M	S	S	M	M	-	S	-	-	S	-	-	M	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

#### Detailed syllabus:

**Review of Probability and Statistics:** Gaussian Distribution, Poisson Distribution, Exponential Family Distributions, Conditional Multivariate Gaussian, Marginalization, Conjugacy

**Introduction:** Bayesian Learning – Bayes theorem, Bayes Optimal Classifier

**Regression:** Multi-Variate Linear Regression, Method of Least Squares, Model Selection, Evaluation of Regression Models, Regularization (Lasso & Ridge), Alternating Least Squares for solving recommendation systems

**Classification:** Likelihood vs Loss minimization for classifiers, Naïve Bayes, Multi-Class Logistic Regression, Regularization in Bayesian and Loss Minimization Scenarios, Gradient Descent vs Quasi-Newton methods for scaling the optimization

**Unsupervised Learning:** K-Means Clustering, Expectation Maximization for missing value problems, EM for clustering, EM for Gaussian Mixture Models, Spectral Clustering

**PCA for feature selection,** SVD, Spectral Clustering for dimensionality reduction, EM for PLSA

**Kernal Machines:** Support Vector Regression and Classification, hard-margin vs soft-margin classifiers, dual sparsity in SVMs, Multi-Class SVM, large-scale SVM learning

**Computational Learning Theory** – Sample Complexity for Finite Hypothesis spaces, Sample Complexity for Infinite Hypothesis spaces, The Mistake Bound Model of Learning;

**Sequence Labelling:** Hidden Markov Models, Parameter Estimation using EM, Conditional Random Fields

**Artificial Neural Networks** – Perceptrons, Multilayer networks, Activation Functions, Derivation of backpropagation rule-Backpropagation Algorithm- Dropout, Batch Normalization, Layer Norm, Recurrent Neural Network, Vanishing and Exploding Gradients and LSTM for Sequence Labelling

**Word Embedding Methods:** Word2Vec, Glove, FastText, Transformers

#### **Reading List:**

1. C Bishop – Pattern Recognition and Machine Learning – Springer, 2006.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville – Deep Learning, The MIT Press Cambridge, Massachusetts, London, England, 2016
3. Tom.M.Mitchell, Machine Learning, McGraw Hill International Edition, 1997.

<b>CS3012</b>	<b>Machine Learning Lab</b>	<b>PCC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	-----------------------------	------------	------------------	-----------------

**Pre-requisites:**

- i. Modelling and Optimization Techniques (EC2511)
- ii. Probability, Statistics and Stochastic Processes (MA2031)
- iii. Differential and Integral Calculus (MA1011)
- iv. Matrices and Differential Equations (MA1021)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Design and implement algorithm using least means square, collaborative filtering
<b>CO2</b>	Design and implement classifiers using Naïve Bayes, Logistic Regression, SVM, Neural Networks
<b>CO3</b>	Design and implement EM for clustering, GMM, Sequence Learning.
<b>CO4</b>	Design and implement feature selection using PCA, SVD
<b>CO5</b>	Design and implement sequence labellers using HMM, CRF and LSTM

**Course Articulation Matrix**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	M	S	S	M	M	-	S	-	-	S	-	-	M	S	M
<b>CO2</b>	M	S	M	M	M	-	S	-	-	S	-	-	M	S	M
<b>CO3</b>	M	S	S	M	M	-	S	-	-	S	-	-	S	S	S
<b>CO4</b>	M	M	S	M	M	S	M	-	-	S	S	-	M	M	M
<b>CO5</b>	M	M	S	M	M	S	M	-	-	S	S	-	M	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Case Studies:** Spam detection, stock-price prediction, whether forecast, face recognition, digit recognition,

**Regression:** Least Squares for regression, evaluation and model selection for regression, ALS for collaborative filtering, SVR

**Classification:** Design of Classifiers using Naïve Bayes, Logistic Regression, SVM and Neural Networks, Implementation of backpropagation for softmax, neural networks and weak supervision, Evaluation of Classifiers, Regularization, Error analysis through matplotlib, correlation plots, t-sne, heatmaps

Design of Classifiers for Semi-Supervised, Transfer Learning, EM for GMM,

**Feature Selection:** Feature Selection using PCA, Dimension reduction using SVD, Document Modelling through PLSA

**Sequence Labelling:** Design a sequence labellers using HMM, CRF, RNN and LSTM

Design a Convolutional network for handwritten character recognition problem for A to Z and 0 to 9 digits.

**Reading List:**

1. Tom M. Mitchell, "Machine Learning", Mc Graw Hill, Indian Edition, 2017.
2. Ethem Alpaydin, "Introduction to Machine Learning", MIT Press, Fourth Edition, 2020.
3. Sebastain Raschka, "Python Machine Learning", Packt publishing (open source).
4. M.A. Bramer, Logic Programming with PROLOG, Springer

<b>CS3021</b>	<b>Software Engineering</b>	<b>PCC</b>	<b>2-0-0</b>	<b>2 Credits</b>
---------------	-----------------------------	------------	--------------	------------------

**Pre-Requisites:**

- i. Software Development (CS2061)
- ii. Software Development Lab (CS2062)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Analyze various software engineering models for software project design and choose the most appropriate one for the problem. (Analyze)
<b>CO2</b>	Design and validate SRS documents to develop a quality project. (Create)
<b>CO3</b>	Evaluate the effectiveness of an organization’s software development/ implementation practices, suggest improvements, and define a process improvement strategy. (Apply)
<b>CO4</b>	Draw UML diagrams in order to forward and reverse engineer the complex software projects. (Create)
<b>CO5</b>	Apply manual and automated testing methods to verify and validate the software. (Apply)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	M	L	M		L	M	L		M	M	M		S	S	
<b>CO2</b>	M	L	M			M	M		M	M	L		S	S	
<b>CO3</b>	M	L	M			M	M			L	M		S	S	
<b>CO4</b>	M	L	M		M	M	M		M	M	L		S	S	
<b>CO5</b>	M	L	M		M	M	M		M	M	L		S	S	

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Introduction to Software Engineering**, Software Process, Software life cycle models: Waterfall Model and its Extensions, Rapid Application Development (RAD), Spiral Model, Introduction to Agility, Agile process, Extreme programming, XP Process. Storyboarding.

**Software Requirements:** Functional and Non-Functional, User requirements, System requirements, Software Requirements Document, Attributes of bad SRS documents, Requirement Engineering Process: Feasibility Studies, Requirements elicitation and analysis, requirements validation, requirements management, Classical analysis: Structured system Analysis, Petri Nets, Data Dictionary.

**Design process:** Design Concepts, Design Model, Design Heuristic, Architectural Design, Architectural styles, Architectural Design, Architectural Mapping using Data Flow.

**User Interface Design:** Interface analysis, Interface Design, Component level Design: Designing Class based components, traditional Components.

**Software Design:** Approaches to software design: function-oriented design, object-oriented design. Object Modelling Using UML: Basic object orientation Concepts, Use case Model, Class diagram, Interaction diagrams, Activity Diagram, state chart Diagram, Component and Deployment diagrams. System design examples.

**Software testing:** White box testing, black box testing, Regression Testing, Unit Testing, Integration Testing, Validation Testing, System Testing and Debugging. Test driven development, Junit, Test Automation, Bug life cycle and Tracking.

**Software Implementation Techniques:** Coding practices, Refactoring, Maintenance and Reengineering, BPR model, Reengineering process model, Reverse and Forward Engineering. Version control, CI/CD.

**Software Project Management:** LOC Estimation, FP Based Estimation, COCOMO I & II Model. Project Scheduling, Earned Value Analysis Planning, Project Plan, Planning Process, RFP Risk Management, Identification, Projection, Risk Management, Risk Identification, Risk Mitigation, RMMM Plan, CASE TOOLS.

### **Reading List:**

1. Roger S. Pressman, Software Engineering, A Practitioner's approach, McGraw Hill, 8th edition, 2014.
2. Ian Sommerville, "Software Engineering", 7th edition, Pearson education.7/e, 2005.
3. Rajib Mall, "Fundamentals of Software Engineering", Third Edition, PHI Publication, 2009.
4. Timothy C Lethbridge, "Object-Oriented Software Engineering, Practical software development using UML and Java", McGraw-Hill Education, 2 nd edition, 2004.

5. Jeevan Kumar, "Grokking the Object-Oriented Design Interview" <https://akshay-iyangar.github.io/system-design/>
6. Kent Beck, Cynthia Andres, "Extreme Programming Explained: Embrace Change", 2nd edition, Addison Wesley, 2004.

<b>CS3022</b>	<b>Software Engineering Lab</b>	<b>PCC</b>	<b>0-0-2</b>	<b>1 Credit</b>
---------------	---------------------------------	------------	--------------	-----------------

**Pre-Requisites:**

- i. Software Development (CS2061)
- ii. Software Development Lab (CS2062)

<b>CO1</b>	Prepare a comprehensive Software Requirements Specification (SRS) document adhering to software engineering standards.(Create)
<b>CO2</b>	Draw different types of UML diagrams in order to forward engineer the complex software projects. (Create)
<b>CO3</b>	Develop the code using an optimistic approach with Test-Driven Development (TDD), and perform both unit testing and integration testing.(Create)
<b>CO4</b>	Design and execute test suites and perform automated testing using relevant testing frameworks and tools.(Create)

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	M	L	L	L	M			M	L	L	M		L	S	
<b>CO2</b>	L	M	L		M			M	M	M	L		M	M	
<b>CO3</b>	L	M	L		L	M		M		L	M		L	S	
<b>CO4</b>	M	M	M		L	S	M		L	L	M		M	L	

S: Strong correlation, M: Medium correlation, L: Low correlation

Choose a small-scale real-life design problem/challenge and incrementally build the solution following the UX Design methodology. You are advised to carry out the distributed product/service/solution development using some tool such as JIRA community edition or any other similar tool. Students should follow the agile development processes, team/role definitions, scrum method, pair programming and other techniques learned in this course for the development of the solution design.

**List of Experiments**

1. Identify the requirements from the problem statement. Conduct feasibility study and validate the requirements.
2. Prepare SRS document

3. Design usecase diagrams by capturing the usecase from the problem statement.
4. Model Class, Activity, Sequence and collaboration diagram and apply forward engineering.
5. Develop the code using test driven development and write J-unit testcases.
6. Develop test suites and perform automation testing.

**Reading List:**

1. Roger S. Pressman, Software Engineering, A Practitioner's approach, McGraw Hill, 8th edition, 2014.
2. Ian Sommerville, "Software Engineering", 7th edition, Pearson education.7/e, 2005.
3. Rajib Mall, "Fundamentals of Software Engineering", Third Edition, PHI Publication, 2009.
4. Timothy C Lethbridge, "Object-Oriented Software Engineering, Practical software development using UML and Java", McGraw-Hill Education, 2 nd edition, 2004.
5. Jeevan Kumar, "Grokking the Object-Oriented Design Interview" <https://akshay-iyengar.github.io/system-design/>
6. Kent Beck, Cynthia Andres, "Extreme Programming Explained: Embrace Change", 2nd edition, Addison Wesley, 2004.

<b>CS3031</b>	<b>Computer Networks</b>	<b>PCC</b>	<b>2 – 1 – 0</b>	<b>3 Credits</b>
---------------	--------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Operating Systems (CS2021)
- ii. Design and Analysis of Algorithms (CS2031)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Assess the functionalities of OSI and TCP/IP models (Analyze)
<b>CO2</b>	Analyze MAC layer protocols and LAN technologies (Analyze)
<b>CO3</b>	Design applications using internet protocols (Apply)
<b>CO4</b>	Construct routing and congestion control algorithms (Apply)
<b>CO5</b>	Construct application layer protocols (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	S	S	M									S	S	M
<b>CO2</b>	S	M	L										S	S	M
<b>CO3</b>	S	M	L										S	S	M
<b>CO4</b>	S	M	L										S	S	M
<b>CO5</b>	S	S	S	M									S	S	S

**Detailed syllabus:**

Introduction – Network architecture - Protocol implementation issues - Network design. Reference models - The OSI Reference Model - the TCP/IP Model - A Comparison of the OSI and TCP/IP Models

Datalink Layer- Framing and error detection and correction - Sliding window protocol - Media access control - Ethernet - CSMA - Token ring - Switching - Spanning tree protocol - LANs.

Network layer – Network layer design issues - Internet protocol (IP) – Addressing and forwarding – ARP protocols – Network address translation (NAT) - Routing algorithms - Congestion control algorithms – Internetworking.

Transport layer – User datagram protocol (UDP) - Transmission control protocol (TCP) – TCP connection management - TCP congestion control – TCP versions – sliding window implantation in TCP.

Application layer - Domain name service (DNS) – Simple mail transfer protocol – File transfer protocol - World wide web - Hypertext transfer protocol – Peer-to-peer networking.

**Reading List:**

- 1) Larry L Peterson, Bruce S Davis, Computer Networks, 5th Edition, Elsevier, 1012.
- 2) Andrew S. Tanenbaum, David J Wetherall, Computer Networks, 5th Edition, Pearson Edu.
- 3) Forouzan, Behrouz A., and Firouz Mosharraf. "Computer networks: a top-down approach." (2012).

<b>CS3032</b>	<b>Computer Networks Lab</b>	<b>PCC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	------------------------------	------------	------------------	-----------------

**Pre-requisites:**

- i. Design and Analysis of Algorithms (CS2031)
- ii. Design and Analysis of Algorithms Lab (CS2032)
- iii. Operating Systems (CS2021)
- iv. Operating Systems Lab (CS2022)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Develop programs for client-server applications
<b>CO2</b>	Perform packet sniffing and analyze packets in network traffic.
<b>CO3</b>	Implement error-detecting and correcting codes
<b>CO4</b>	Implement Symmetric and Asymmetric Cryptographic algorithms
<b>CO5</b>	Analyze the network traffic and simulate the attacks

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	S		M		M					M	M	M	
<b>CO2</b>	M	S	M	M	L	L		M				L	L	M	
<b>CO3</b>	L	L	L	M									L		
<b>CO4</b>	L	L	L	M									L		
<b>CO5</b>	L	L	L	M									L		

v. S: Strong correlation, M: Medium correlation, L: Low correlation

## **List of Experiments**

1. Analyze and compare the performance of Inter-Process Communication (IPC) mechanisms—Named Pipes (FIFO) and Shared Memory—by implementing a client-server model.
2. Implement Bit Stuffing, Checksum, and CRC techniques, and critically evaluate their effectiveness in detecting and correcting transmission errors. Analyze error detection capabilities under different error scenarios and transmission conditions.
3. Implement a program for Stop-and-Wait protocol to model reliable data transmission, handling acknowledgment and retransmission of lost packets.
4. Implement the Sliding Window protocol to simulate efficient and reliable data transmission with flow control and error management in a network environment.
5. Develop a socket-based communication model and critically analyse the behaviour of connection-oriented (TCP) vs. connectionless (UDP) sockets in terms of reliability, speed, and suitability for various applications.
6. Construct a real-time chat application supporting both TCP and UDP protocols, applying socket programming to facilitate two-way client-server communication.
7. Design a subnet calculator that generates forwarding tables with key metrics (subnet mask, broadcast address, no of host per subnet, First Host ID, Last Host ID, subnet address etc.). Evaluate the impact of different subnetting strategies on address utilization and routing efficiency in scalable networks.
8. Implement and compare Distance Vector Routing and Link State Routing algorithms to simulate dynamic routing decisions based on network topology changes. Demonstrate route computation and table updates.
9. Capture and analyze the Dynamic Host Configuration Protocol (DHCP) traffic using network simulation tool and prepare a detailed report on the observed DHCP message exchange process, identifying the roles of each packet in IP address allocation.
10. Design a Secure File Integrity and Password Management System Using Cryptographic Hash Functions.

11. Build a secure file encryption utility that applies DES (in CBC modes) and AES (in CBC modes) to encrypt and decrypt files, while generating cryptographically strong keys and initialization vectors.
12. Implement a two-party Diffie-Hellman key exchange by generating a prime modulus and base, exchanging public values, and deriving the shared secret key.
13. Implement RSA encryption with PKCS#1 v1.5 padding:
  - a) Use secure padding for plaintext
  - b) Encrypt and decrypt messages using public-private key pairs
14. Develop a tool that applies RSA or ECDSA to digitally sign documents and verifies signatures to detect any alterations, ensuring document authenticity and integrity.
15. Build a system that encrypts and decrypts by XORing plaintext with a random key of the same length, shows that decryption perfectly recovers the original message, and demonstrates that the same ciphertext can match many different plaintexts, proving perfect secrecy.
16. Develop an authentication simulation for corporate network where multiple clients request and receive encrypted tickets from the Authentication Server (AS) and Ticket Granting Server (TGS) to access internal services, implementing timestamp checks, session key handling, and replay attack prevention while managing concurrent authentication flows in a network environment.
17. Implement a program that automatically decrypts a shift cipher by estimating the key through English letter frequency analysis, correctly handling case, preserving non-alphabetic characters, and outputting the decrypted text with the detected key.
18. Implement a monitoring module that automatically scans and deciphers simple encrypted traffic (like Caesar or monoalphabetic ciphers) used by malware to avoid plain-text detection.
19. Develop a real-time LAN monitoring tool that captures packets, verifies MAC-IP consistency using ARP mapping, and detects IP spoofing attempts by flagging mismatched pairs and logging alerts for intrusion detection simulation.
20. Implement a program for banking system using RPC, where clients remotely request transactions, account info, and statements from replicated servers with synchronization, consistency checks, and rollback capabilities on transaction failure.

## Reading Lists:

- 1) W. Richard Stevens, UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI, Prentice Hall, 1998
- 2) W. Richard Stevens, UNIX Network Programming, Volume 2, Second Edition: Interprocess Communications, Prentice Hall, 1999
- 3) W. Richard Stevens, Stephen Rago, Advanced Programming in the UNIX Environment, Pearson Education, Second Edition.
- 4) A. Menezes, P. Van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRCPress, 2004.
- 5) Swarup Bhunia and Mark Tehranipoor, Hardware Security: A Hands-on Learning Approach, Elsevier, 2019.
- 6) William Stallings and Lawrie Brown, Computer Security: Principles and Practice, Pearson, 2017.
- 7) Trent Jaeger, Operating System Security, Morgan & Claypool Publishers, 2009.
- 8) A. Menezes, P. Van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRCPress, 2004.
- 9) Swarup Bhunia and Mark Tehranipoor, Hardware Security: A Hands-on Learning Approach, Elsevier, 2019.
- 10) William Stallings and Lawrie Brown, Computer Security: Principles and Practice, Pearson, 2017.
- 11) Trent Jaeger, Operating System Security, Morgan & Claypool Publishers, 2009.

<b>CS3051</b>	<b>Distributed Computing</b>	<b>PCC</b>	<b>2-1-0</b>	<b>3 Credits</b>
---------------	------------------------------	------------	--------------	------------------

Students of this course upon completion will learn how to solve standard distributed computing problems. Students will also learn consensus algorithms. Further algorithms for global knowledge, recording distributed snapshots, distributed mutual exclusion with an emphasis to reduce communication cost are also studied.

**Pre-requisites:**

- i. Design and Analysis of Algorithms (CS2031)
- ii. Computer Networks (CS3031)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Construct efficient distributed algorithms for standard graph-theoretic problems with an analysis of their time, space and communication requirements (Apply)
<b>CO2</b>	Constructing efficient distributed algorithms for achieving stable properties (Apply)
<b>CO3</b>	Analyze the properties of various algorithms for fault tolerance/ consensus (Analyze)
<b>CO4</b>	Construct and analyze algorithms for solving distributed mutual exclusion. (Apply)
<b>CO5</b>	Analyze the properties of distributed transaction processing/ concurrency algorithms (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	S	M										S	M	L
<b>CO2</b>	S	M	L										S	L	L
<b>CO3</b>	S	S	M										S	M	L
<b>CO4</b>	S	S	M										S	M	L
<b>CO5</b>	S	S	M										S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

Taxonomy of distributed systems, Failure Detection, Clocks and Time, Logical Time, Distributed Consistency Models,

Leader election in rings. Basic computational tasks in general synchronous networks: leader election. Breadth-first search. Shortest Path Algorithms, Floyd-Warshall's Algorithm, Minimal Spanning Trees, Maximal Independent Sets,

Stable property detection. Distributed termination. MultiCast Models, Distributed snapshot recording. Deadlock detection. Asynchronous shared-memory systems. The mutual exclusion problem, Distributed Mutual exclusion algorithms.

Fault-tolerant consensus, Link failures: the two generals' problem. Process failures (stopping, Byzantine). Algorithms for agreement with stopping and Byzantine failures. PBFT, Paxos, zookeeper, RAFT, A case study of Bitcoin consensus

Distributed Transaction Processing and Concurrency Control, 2PC, 3PC vs View Stamped Replication, 2PL, Distributed Deadlocks

Distributed Hash Tables, Key-Value Stores (Cassandra)

## **Reading List:**

- 1) A. Kshemkalyani and M. Singhal, Distributed Computing: Principles, Algorithms, and Systems Cambridge University Press 2008 ISBN: 978-0-521-87634-6
- 2) Lynch, Nancy. Distributed Algorithms. Burlington, MA: Morgan Kaufmann, 1996. ISBN:9781558603486.
- 3) Michel Raynal Distributed Algorithms for Message-Passing Systems ISBN 978-3-642-38122-5 © Springer-Verlag Berlin Heidelberg 2013
- 4) A Guided Tour on the Theory and Practice of State Machine Replication, Alysson Neves Bessani and Eduardo Alchieri Book Chapter
- 5) Relevant Research articles shared by the instructor.

<b>CS3052</b>	<b>Distributed Computing lab</b>	<b>PCC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	----------------------------------	------------	------------------	-----------------

**Pre-requisites:**

- i. Design and Analysis of Algorithms (CS2031)
- ii. Design and Analysis of Algorithms Lab (CS2032)
- iii. Distributed Computing (CS3051)

**Course Outcomes:** At the end of the course, student will be able to:

<b>CO1</b>	Design and implement Distributed Graph Algorithms
<b>CO2</b>	Design and implement Distributed Leadership and Synchronization algorithms
<b>CO3</b>	Design and implement Distributed File System
<b>CO4</b>	Design and implement Distributed Consensus Algorithms

**Course Articulation Matrix**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	M	L	L	-	M	-	-	-	S	-	S	M	S	M	-
<b>CO2</b>	-	L	L	M	L	S	-	-	S	-	S	M	-	-	L
<b>CO3</b>	L		L	L	L	S	S	S	L	M	L	L	L	M	L
<b>CO4</b>	-	L	M	S	L	M	S	S	M	M	L	L	-	-	S

**Topics:**

Distributed Graph Algorithms: MSP, Floyd Warshall’s All-Pairs Shortest Path, Single Source shortest Path, MIS

Distributed Leader Election Algorithms, Distributed Process Synchronization

DFS Implementation, Distributed Shared Memory

Distributed Consensus: Paxos, RAFT, Viewstamped Replication, ZAB, BFT(PBFT, Tendermint, HotStuff, Algorand)

Gossip-Based Consensus, Eventual Consistency,

## Consistency in Blockchains, Merkle Trees, Merkle Tries

### Reading List:

- 1) GNU-MP Manual
- 2) A. Kshemkalyani and M. Singhal, Distributed Computing: Principles, Algorithms, and Systems  
Cambridge University Press 2008 ISBN: 978-0-521-87634-6
- 3) Lynch, Nancy. Distributed Algorithms. Burlington, MA: Morgan Kaufmann, 1996.  
ISBN:9781558603486.
- 4) Michel Raynal Distributed Algorithms for Message-Passing Systems ISBN 978-3-642-38122-5  
© Springer-Verlag Berlin Heidelberg 2013
- 5) A Guided Tour on the Theory and Practice of State Machine Replication, Alysson Neves  
Bessani and Eduardo Alchieri Book Chapter
- 6) Relevant Research articles shared by the instructor.

<b>CS3061</b>	<b>Cryptography and Network Security</b>	<b>PCC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--	------------	------------------	------------------

**Pre-requisites:**

- i) Design and Analysis of Algorithms (CS2031)
- ii) Computer Networks (CS3031)

**Course Outcomes:** At the end of the course, student will be able to:

<b>CO1</b>	Assess and exploit vulnerabilities in software and hardware platforms. (Analyze)
<b>CO2</b>	Assess the systems for certain form of vulnerabilities. (Analyze)
<b>CO3</b>	Construct software and hardware for security evaluation of well-known attacks. (Apply)
<b>CO4</b>	Construct and apply software validation and verification techniques to test security vulnerabilities. (Apply)
<b>CO5</b>	Construct case studies to think like an attacker in order to expose security vulnerabilities in software systems. (Apply)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	S	S	M	-	-	-	-	-	-	-	-	S	S	S
<b>CO2</b>	S	S	S	M	-	-	-	-	-	-	-	-	S	S	S
<b>CO3</b>	S	M	L	-	-	-	-	-	-	-	-	-	S	S	M
<b>CO4</b>	S	M	L	-	-	-	-	-	-	-	-	-	S	S	M
<b>CO5</b>	S	M	L	-	-	-	-	-	-	-	-	-	S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

### **Detailed syllabus:**

**Basic Cryptography-** Security Services, Security Attacks, Basic cryptography methods, Classical Encryption Techniques, Substitution Ciphers, Transposition Ciphers, one time pad, Cryptanalysis Security attacks, Symmetric cryptography algorithms- DES, 3DES, AES, Public-key cryptography. Number Theory- Introduction to number theory – Modular Arithmetic; Finite fields; Number theory properties – Primality testing; Fermat’s and Euler’s theorem; Chinese remainder theorem; Integer factorization; discrete logarithm. Public Key Cryptography: RSA, ECC, Generalized ElGamal

**Public Key Cryptosystem,** Key Management and Distribution: Symmetric Key Distribution, Diffie Hellman key exchange, Distribution of Public Keys, X.509 Certificates Functionalities of entity, content authentication; message digests and hashing schemes; Key management and Distribution- Certificate authorities; PKI; MAC; Hashing; Digital Signatures Authentication protocols; Digital Signature Standard.

**Network Security:** IP Security, Transport-Level Security, Electronic Mail Security, Firewalls- Firewall Properties; Design of firewalls; VPN’s; Filtering; Intrusion detection system.

### **Reading List:**

- 1) A. Menezes, P. Van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, 2004.
- 2) William Stallings and Lawrie Brown, Computer Security: Principles and Practice, Pearson, 2017.
- 3) A Graduate Course in Applied Cryptography by Dan Boneh and Victor Shoup
- 4) The Joy of Cryptography by Mike Rosulek, School of Electrical Engineering & Computer Science Oregon State University, Corvallis, Oregon, USA
- 7) Introduction to Cryptography, Principles and Applications by Hans Delfs and Helmut Knebl.

<b>CS3071</b>	<b>Theory and Design of Programming Languages</b>	<b>PCC</b>	<b>2-1-0</b>	<b>3 Credits</b>
---------------	---	------------	--------------	------------------

This course aims to establish a broad framework in which to formulate and analyze the design aspects of programming languages. Students of this course will be exposed to multiple programming language design paradigms, their underlying principles, representative programming languages, their constructs and programming techniques.

**Pre-requisites:**

- i. Theory of Computation (CS2041)
- ii. Introduction to Computational Problem Solving (CS1011)
- iii. Object Oriented Design and Programming (CS1061)
- iv. Introduction to Computational Problem-Solving Lab (CS1012)
- v. Object Oriented Design and Programming Lab (CS1062)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Demonstrate a comprehension of the concepts found across different programming languages using a precise single math framework. (Apply)
<b>CO2</b>	Prove the properties of programming languages using structural induction with a comprehension of abstract syntax, binding, and scope of identifiers. (Analyze)
<b>CO3</b>	Construct precise math formalisms for specifying the type-system and the runtime behavior of a program. (Analyze)
<b>CO4</b>	Prove that a language is safe with respect to types. (Analyze)
<b>CO5</b>	Construct programs across languages from multiple design paradigms with a comprehension of their underlying design principles, constructs, and a sensible integration of multiple features of the language. (Analyze)

**Course Articulation Matrix**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	L										S	M	
<b>CO2</b>	S	S	M										S	M	
<b>CO3</b>	S	S	M										S	M	

<b>CO4</b>	S	S	M										S	M	
<b>CO5</b>	S	S	M		S			S	S	S		S	S	S	L

S: Strong correlation, M: Medium correlation, L: Low correlation

### Detailed Syllabus:

**Syntactic Objects:** Abstract Syntax Trees, Abstract Binding Trees, Inductive definitions, Hypothetical and general judgements: Derivability, admissibility.

**Static and dynamic semantics:** Type safety, Binding, Binding within Scope, Scope rules, Generic Programming: inductive and co-inductive types, Polymorphism, genericity, Existential types.

**Control abstractions:** Data abstraction, Exceptions and control data. Programming with continuations, Dynamic Classification, Lazy evaluation.

**Higher-order total computation:** Gödel's T, Higher-order programming in T, Parametricity, verification of ADTs, Higher-order partial computation: Plotkin's PCF.

**Dynamic and Unityped Languages:** FPC, Dynamic Languages: DPCF, Dynamic dispatch,  $\lambda$ -calculus and DPCF.

**Imperative programming Paradigm:** Modernized Algol constructs and programming techniques

**Functional Programming Paradigm:** Haskell and ML constructs and programming techniques, **Concurrent Programming Paradigm:** Parallelism: Work & span, Brent's theorem.

**Logic Programming paradigm:** Prolog constructs, and programming techniques

### Indicative Tutorial Exercises:

1. Familiarization with the syntactic constructs of Modernized Algol.
2. Exercises involving imperative programming languages like Modernized Algol.
3. Familiarization with Concurrent programming constructs.

4. Case Studies in Concurrent programming.
5. Familiarization with the syntactic constructs of Haskell.
6. Case Studies in Haskell.
7. Familiarization with ML constructs.
8. Case Studies in ML.
9. Familiarization with Prolog constructs.
10. Case Studies in Prolog

**Reading List:**

- 1) Robert Harper, Practical Foundations for Programming Languages, Cambridge University Press, Second Edition, 2016.
- 2) Michael L. Scott, "Programming Language Pragmatics", Fourth Edition, Morgan Kaufmann, 2016.

## II Year II Semester Department Electives Syllabus

<b>CS2402</b>	<b>Data Processing and Visualization Lab</b>	<b>DEC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	--	------------	------------------	-----------------

### Pre-requisites:

- i) **Introduction to Computational Problem Solving (CS1011)**
- ii) **Introduction to Computational Problem-Solving Lab (CS1012)**

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Construct and implement multidimensional array operations using NumPy.
<b>CO2</b>	Construct and implement robust data manipulation and analysis workflows using pandas.
<b>CO3</b>	Implement data input and output workflows and Data Manipulation and Transformation Techniques using pandas.
<b>CO4</b>	Create scatter plots and static and interactive visualizations with matplotlib.

### Course Articulation Matrix:

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	L		S				M			L	L	S	S
<b>CO2</b>	S	M	L		S				M			L	L	S	S
<b>CO3</b>	S	M	L		S				M			L	L	S	S
<b>CO4</b>	S	M	L		S				M			L	L	S	L

S: Strong correlation, M: Medium correlation, L: Low correlation

### Tentative List of Laboratory Exercises:

1. Implement creating arrays, handling various data types, performing operations between arrays and scalars, and utilizing indexing methods.
2. Implement file I/O operations, including storing arrays in binary and text formats, and linear algebra.

3. Implement handling Series and DataFrame data structures, handling Index objects, performing essential functionalities.
4. Implement handling and filling missing data, hierarchical indexing, reordering and sorting levels, and summarizing statistics by level.
5. Implement reading and writing data in various text formats, handling delimited data manually, and working with JSON, XML, and HTML data for web scraping.
6. Implement reading and writing data in various text formats, handling delimited data manually, and working with JSON, XML, and HTML data for web scraping.
7. Implement combining and merging datasets with joins, merging on indices, and concatenating along axes.
8. Implement data transformations such as removing duplicates, applying functions and mappings, replacing values, and renaming axis indexes.
9. Implement string manipulation using methods, regular expressions, and vectorized string functions in pandas
10. Implement creation and customization of figures, subplots, and various types of plots including line plots, bar plots, histograms, density plots, and scatter plots.
11. Case study/Mini project

### **Reading List:**

1. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython, 2nd edition, O'Reilly Media, 2017
2. A. B. Downey, Think Python, 2e: How to Think Like a Computer Scientist, O'Reilly, 2015.
3. Z. Shaw, LEARN PYTHON 3 THE HARD WAY, Addison-Wesley, 2017.
4. Arockia Mary P, Problem Solving and Python Programming, Shanlax Publications, 2021.
5. <https://docs.python.org/3/tutorial/index.html>.

<b>CS2412</b>	<b>Modelling and Optimization Lab</b>	<b>DEC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	---------------------------------------	------------	------------------	-----------------

**Pre-requisites:**

- i. Modelling and Optimization Theory (EC2511)
- ii. Differential and Integral Calculus (MA1011)
- iii. Discrete Event Simulation (CS2011)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Design and Implement Linear Programming Algorithms
<b>CO2</b>	Apply Linear Programming paradigm to allocation problems
<b>CO3</b>	Design and implement Quadratic Programming Algorithm

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	L	L									S	S	S
<b>CO2</b>	S	M	L	L									S	S	S
<b>CO3</b>	S	M	L	L									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Tentative List of Laboratory Exercises:**

1. Write a program to implement simplex method for solving linear programming problems
2. Write a program to implement tableau method for solving linear programming problems. Compare its performance with that of simplex method.
3. Implement two-phase method and big-M methods for linear programming
4. Write programs for solving the linear programming problem using Bland's rule and perturbation technique
5. Implement bounded variable Simplex method for bound constrained LPP
6. Implement column generation method for cutting stock problem
7. Implement Simplex method using product form of inverse
8. Write program for LU decomposition-based solution for LPP
9. Implement an LPP based solution for solving the Network flow problem
10. Implement an LPP based solution for solving the shortest path problem

11. Implement ellipsoid method for solving LPPs
12. Implement primal-dual based interior point method for solving LPPs
13. Implement predictor-corrector method for solving LPPs
14. Write a program to plot the contours of given quadratic function. Plot the gradient vectors
15. Solve a constrained non-linear program. Draw the contours of the objective function and constraints. Obtain the solution for the same.
16. Find solutions to a polynomial using bisection method and Newton-Raphson method
17. Implement conjugate gradient for solving a given quadratic optimization problem
18. Implement Quasi Newton's method using BFGS method.
19. Case study/Mini project

### **Reading List:**

1. Linear and Nonlinear Optimization SECOND EDITION Igor Griva, Stephen G. Nash, Ariela Sofer Siam Publishers
2. Convex Optimization by Stephen Boyd and Lieven Vandenberghe
3. [https://web.stanford.edu/~boyd/cvxbook/bv\\_cvxbook.pdf](https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf)
4. <https://people.duke.edu/~ccc14/sta-663-2018/labs/>

<b>CS2422</b>	<b>Mobile Application Development Lab</b>	<b>DEC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	---	------------	------------------	-----------------

**Pre-requisites:**

- i) **Introduction to Computational Problem Solving (CS1011)**
- ii) **Introduction to Computational Problem-Solving Lab (CS1012)**

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Create a mobile app with user interaction components such as text, image, audio, video
<b>CO2</b>	Create a mobile app which uses voice inputs to perform tasks
<b>CO3</b>	Create a mobile app of societal relevance

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M		S				M	M			S	S	S
<b>CO2</b>	S	M	M		S				M	M			S	S	S
<b>CO3</b>	S	M	M		S				M	M			S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Tentative List of Experiments:**

1. Install SDK for Mobile application and build a test project to verify the configuration of the installed tools
2. Develop an App to display interactive icons. Include user input functionality and navigation in the App. Support to image capture through camera and play audio and video content
3. Develop an App where user can be navigated from one App to another App to complete a task. Example: From the developed App, user can go to Maps or call a phone number. Experiment with relevant practical layouts.
4. Develop an App integrating with standard apps available on mobile. Example: App integrated with Maps
5. Develop an App to convert speech to text. Customize this App to support Web and YouTube search in the student's native Indian language
6. Develop an App for a task requiring SMS OTP Verification

7. Develop a GPS enabled application
8. Develop a capstone project of student's choice to help elderly people or useful to the local people

**Reading List:**

1. Dawn Griffiths and David Griffiths, Head First Android Development: A Brain-Friendly Guide, O'Reilly, 2017
2. John Horton, Android Programming for Beginners, Second Edition, Packt Publishing, 2018
3. Matt Neuburg, iOS 15 Programming Fundamentals with Swift, O'Reilly Media, 2021
4. Matthew Mathias and John Gallagher, Swift Programming: The Big Nerd Ranch Guide, 1st edition, Big Nerd Ranch Guides, 2015
5. <https://developer.android.com/develop/ui/compose/tutorial>
6. <https://cs193p.sites.stanford.edu/2023>
7. <https://www.coursera.org/learn/introduction-to-android-mobile-application-development>
8. <https://www.udacity.com/course/developing-android-apps-with-kotlin--ud9012>
9. Case study/Mini project.

<b>CS2432</b>	<b>Software Coding Standards lab</b>	<b>DEC</b>	<b>0 – 0 – 2</b>	<b>1 Credit</b>
---------------	--------------------------------------	------------	------------------	-----------------

**Pre-requisites:**

- i) **Introduction to Computational Problem Solving (CS1011)**
- ii) **Introduction to Computational Problem-Solving Lab (CS1012)**

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Build code modules by following coding standards and best-practices with respect to the design, implementation, integration, testing and production support.
<b>CO2</b>	Demonstrate effective use of development, testing, debugging, versioning, documentation and deployment tools while developing software.
<b>CO3</b>	Build code modules that are readable, maintainable and reliable.

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	M		S				M			L	S	M	M
<b>CO2</b>	S	M	M		S				M			L	S	M	M
<b>CO3</b>	S	M	M		S				M			L	S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Tentative List of Laboratory Exercises:**

1. Naming standards for files, variables, functions and other user-defined constructs in a program.
2. Commenting standards and auto-generation of documentation from comments.
3. Effective usage of features in Integrated Development Environments.
4. Learning to Explore and self-learn using help files, documentation and man pages.
5. Usage of Versioning repositories: Git
6. Adopting Modular programming practices instead of monolithic coding.
7. Usage of appropriate exceptions and error handling.

8. Claiming and releasing resources such as memory, files and connections judiciously.
9. Avoiding memory leaks and dangling references.
10. Usage of appropriate parameter passing mechanisms.
11. Usage of suitable access controls and visibility specifiers for programming constructs.
12. Continuous integration and testing: Early identification of bugs in code.
13. Using debuggers such as GDB effectively and systematic debugging.
14. Pair programming and Code-walkthroughs as a form of static analysis of the code to uncover errors.
15. Multiple developers working on the same code components using suitable conflict resolution strategies in the versioning repositories.
16. Best practices while creating software libraries such as DLLs.
17. Coding assertions and enabling conditional logging to debug programs.
18. Analysis of log files and management of log files in production.
19. Working with different environments such as development, Test and Production and promotion or demotion of code modules from one to the other.
20. Handling production support issues, understanding severity of issues and responding to SLA requirements.
21. Familiarization with Code Profilers
22. Case study/Mini project

### **Reading List:**

1. David Thomas, Andrew Hunt, The Pragmatic Programmer, 2nd edition, Addison-Wesley, 2019.
2. Robert C. Martin, Matthew Heusser, The Clean Coder, First edition, Pearson Education 2012.

III Year I Semester Department Electives Syllabus

<b>CS3401</b>	<b>Data Warehousing and Data Mining</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---	------------	------------------	------------------

**Pre-requisites:**

- i. Data Structures and Algorithms (CS1051)
- ii. Database Management Systems (CS2051)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Understand the design and implementation of data warehouses.
<b>CO2</b>	Explore OLAP, dimensional modeling, and data mart architecture.
<b>CO3</b>	Apply Business Intelligence tools for analytical reporting and visualization.
<b>CO4</b>	Integrate predictive analytics and performance management into BI systems.

**Course Articulation Matrix:**

PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO	PSO	PSO
<b>CO1</b>	S	S	S	M	M	L							S	S	S
<b>CO2</b>	S	S	S	M	M	L							S	S	S
<b>CO3</b>	S	S	S	M	M	L							S	S	S
<b>CO4</b>	S	S	S	M	M	L							S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Introduction:** Operational Databases, Data Warehouses, Building Data Warehouses, Using Data Warehouses

**Star Schema, Creating Facts and Dimensions:** Count Function, Average in the Fact, Outer Join, Creating Temporary Dimension Tables, Creating Temporary Tables in the Operational Database

**Hierarchies:** Hierarchy vs. Non-hierarchy, Hierarchy Versus Multiple Independent Dimensions, Separate vs. Combined Dimension, Linked Dimensions, Hierarchy Design Considerations, Bridge Tables, Temporal Data Warehousing Determinant Dimensions, Pivoted Fact Table, Junk Dimensions: Non-junk Dimension, Simple Join Queries, Nested Queries Dimension Keys, One- Attribute Dimensions

**Multi-Fact Star Schemas:** Multi-Fact or Single Fact with Multiple Fact Measure, Mutually Exclusive Star Schemas, Different Granularity Multi-Fact, Slicing a Fact, Multi-Input Operational Databases

Data Warehousing Granularity and Levels of Aggregation, Facts Without Fact Measures, Star Schemas with No Aggregation, Relationship Between Transactions and Fact Measures, Levels of Aggregations, Hierarchy and Multi-Fact, Lowest-Level Star Schemas, Higher-Level Star Schema, Active Data Warehousing: Incremental Updates, Data Warehousing Schema Evolution

Online Analytical Processing (OLAP) Aggregate Functions, Cube and Rollup, Ranking, Cumulative and Moving Aggregate, Exploring Dirty Data, Exploring the Extended Fact Table, Pre-and Post-Data Warehousing, Data Analytics for Data Warehousing.

**Introduction to Business Intelligence, BI Infrastructure and Tools:** BI Platforms and Ecosystem, Data Integration: ETL Tools and Techniques, Data Quality and Master Data Management, Overview of BI Tools (e.g., Power BI, Tableau, QlikView, SAP BI) BI Reporting and Querying, Data Visualization and Dashboards, Predictive and Advanced Analytics, Business Performance Management

#### **Reading List:**

1. Taniar, David, and Wenny Rahayu. Data warehousing and analytics: fueling the data engine. Springer Nature, 2022.
2. Inmon, William H., Derek Strauss, and Genia Neushloss. DW 2.0: The architecture for the next generation of data warehousing. Elsevier, 2010.
3. Grossmann, Wilfried, and Stefanie Rinderle-Ma. "Fundamentals of business intelligence". (2015): 348.
4. Sherif, Ahmed. Practical business intelligence. Packt Publishing Ltd, 2016.
5. Van Der Lans, Rick. Data Virtualization for business intelligence systems: revolutionizing data integration for data warehouses. Elsevier, 2012.

#### **References:**

<https://www.coursera.org/learn/business-intelligence-data-warehousing>

<https://www.coursera.org/learn/data-warehousing-business-intelligence#modules>

[BR1505297627176020D02\\_Big\\_data\\_analytics\\_IITKGP.pdf](BR1505297627176020D02_Big_data_analytics_IITKGP.pdf)

[https://archive.nptel.ac.in/content/syllabus\\_pdf/106106361.pdf](https://archive.nptel.ac.in/content/syllabus_pdf/106106361.pdf)

<http://www.cs.iit.edu/~qlavic/cs520/2022-spring/syllabus/>

[https://onlinecourses.nptel.ac.in/noc24\\_cs65/preview](https://onlinecourses.nptel.ac.in/noc24_cs65/preview)

<http://www.facweb.iitkgp.ac.in/~shamik/autumn2011/dwdm/dwdmautumn2011coursedtls.html>

<https://www.careers360.com/university/macquarie-university-sydney/business-intelligence-and-data-analytics-generate-insights-certification-course>

<b>CS3411</b>	<b>Data Science</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Modelling and Optimization Techniques (EC2511)
- ii. Probability, Statistics and Stochastic Processes (MA2031)
- iii. Differential and Integral Calculus (MA1011)
- iv. Matrices and Differential Equations (MA1021)

**Course Outcomes:** At the end of the course students will be able to

<b>CO1</b>	Apply basic machine learning algorithms (Linear Regression, k-Nearest Neighbors (k-NN), k-means, Naive Bayes) for predictive modeling (Apply)
<b>CO2</b>	Analyze efficiency large scale matrix factorizations for different data science scenarios (Analyze)
<b>CO3</b>	Designing Recommender System for different applications. (Analyze)
<b>CO4</b>	Designing feature reduction objectives (Apply)
<b>CO5</b>	Analyze various clustering objectives (Analyze)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	S	-	S	S	-	-	-	-	S	-	-	S	S	L
<b>CO2</b>	-	-	M	-	-	-	-	-	-	M	M	-	S	S	L
<b>CO3</b>	-	-	M	S	-	-	-	S	-	M	-	-	S	M	L
<b>CO4</b>	-	-	-	S	-	S	-	-	M	S	-	-	S	M	L
<b>CO5</b>	-	-	-	S	M	M	M	-	S	M	S	M	S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Basics of Optimization in Data Science:** Linear Programming Objectives, Primal-Dual Methods, Quadratic Programming, Convex Optimization, Gradient Descent, Adaptive Learning Rate, Quasi Newton's method, Constrained Optimization, KKT Conditions for Optimization Problems

Probability theory including random variables, conditional probability, Bayes law, concentration of measure, linear algebra including eigenvalues, norms, elementary spectral graph theory.

**Regression:** Least Squares Objective for Multiple Linear Regression, Gauss-Markov Theorem, Statistical Tests, Weighted Least Squares, Box-Cox Transformation, Polynomial & Spline Regression, Ridge Regression, Bias-Variance Tradeoff, Subset Selection, LASSO, Adaptive LASSO, Elastic Net, Dantzig Selector, SLOPE and sorted penalties.

**Matrix Factorizations used in Data Science:** Cholesky Decomposition and QR Factorization: Concepts, Applications Eigen Vector Decomposition, Solving Large Scale Value Problems using Lanczos Method, Arnoldi's Iteration

**SVD:** Geometric Interpretation, Best Rank-K Approximation using SVD, Power Method for SVD, Efficient methods for SVD, Applications of SVD

**PCA:** PCA learning objective, Application of PCA for dimensionality reduction.

**Recommendation Systems:** Collaborative Filtering using gradient Descent and Alternating Least Squares for recommender systems.

**Classification:** GLM methods for classification, SVM, Naïve Bayes, Evaluation of classification methods – Confusion matrix, Students T-tests and ROC curves, Feature Selection for Classification

**Clustering:** Choosing distance metrics – Different clustering approaches – hierarchical agglomerative clustering, k-means (Lloyd's algorithm), Soft Clustering, Evaluation of clustering algorithms, t-sne embedding

**Link Analysis:** Random Walks, Markov Chains, Stationary Distribution, Metro Polis Hastings Algorithm, Gibbs Sampling, Convergence of Random Walks on Undirected Graphs, Page Rank, HITS, combating spam, personalized page rank.

High Dimensional Space High dimensional sphere, volumes of high dimensional solids, gaussians in high dimension, high dimensional point sets, Johnson-Lindenstrauss theorem.

### **Reading List:**

1. Avrim Blum, John Hopcroft, and Ravindran Kannan: Foundations of Data Science Cambridge University Press, 2020
2. Jianqing Fan, Runze Li, Cun-Hui Zhang, Hui Zou Statistical Foundations of Data Science, CRC Press 2020
3. Jure Leskovec, Anand Rajaraman, Jeff Ullman: Mining of Massive Datasets, Cambridge University Press, 2016

4. G. Strang. Introduction to Linear Algebra, Wellesley-Cambridge Press, Fifth edition, USA, 2016
5. J. Grus, Data science from scratch: first principles with python, 2/e, O'Reilly Media, 2019.
6. L. Igual and S. Seguí, Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications, 1/e, Springer, 2017.
7. C. O'Neil and R. Schutt, Doing Data Science: Straight Talk from the Frontline, 1/e, O'Reilly Media, 2013.
8. Relevant Research articles shared by the instructor.

<b>CS3421</b>	<b>Advanced Database Systems</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	----------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Database Management Systems (CS2051)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Understand the principles, architecture, and design issues of distributed databases, including fragmentation, allocation, and integration.
<b>CO2</b>	Apply query processing techniques, such as query decomposition, optimization, and selectivity estimation, to improve the database performance.
<b>CO3</b>	Analyze concurrency control methods, failure recovery mechanisms, and security issues in distributed databases.
<b>CO4</b>	Evaluate transaction management techniques, including transactional workflows, replication, and real-time transaction systems.
<b>CO5</b>	Develop efficient database solutions using parallel database architectures, query parallelism, load balancing, and database tuning techniques.

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	M										S	M	S
<b>CO2</b>	S	M	M										S	M	S
<b>CO3</b>	S	M	M										S	M	S
<b>CO4</b>	S	M	M										S	M	S
<b>CO5</b>	S	M	M										S	M	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Indexing:** Primary vs Secondary Index, Dense and Sparse Index, Clustered vs Non-clustered Index, B-Tree and B+ Tree, Hash-based Indexing, Multi-level Indexing, Bitmap Indexing, Inverted Index (used in IR), Index Maintenance and Update Strategies.

**Query Processing:** Translating SQL to relational algebra, Evaluation Plans, Cost Estimation, Access Paths (Index scan, Table scan), Pipelining vs Materialization.

**Query Optimization:** Heuristic-based Optimization, Cost-based Optimization, Join Order Optimization, Predicate Pushdown, Use of Indexes, Selectivity Estimation, Query Rewriting and Transformation Rules.

### **Transaction Management:**

**Transactions:** ACID Properties, Transaction States (Active, Partially Committed, Failed, Aborted, Committed), Serializability (Conflict, View).

**Concurrency Control:** Lock-based Protocols (2PL – Two Phase Locking), Timestamp Ordering Protocol, Optimistic Concurrency Control, Deadlock Detection and Prevention, Isolation Levels (Read Uncommitted, Read Committed, Repeatable Read, Serializable).

**Recovery Systems:** Log-based Recovery (Undo/Redo), Checkpointing, Shadow Paging, ARIES (Algorithm for Recovery and Isolation Exploiting Semantics).

**Parallel and Distributed Databases:** Database System Architectures Parallel and Distributed Storage, Parallel and Distributed Query Processing Parallel and Distributed Transaction Processing.

**Database System Architectures:** Shared Memory, Shared Disk, Shared Nothing Architectures, Client-Server and Peer-to-Peer Architectures, Federated Databases.

**Parallel and Distributed Storage:** Data Partitioning (Horizontal, Vertical, Hybrid), Data Replication, Data Fragmentation and Allocation.

**Parallel and Distributed Query Processing:** Intra-query Parallelism (Inter-operator, Intra-operator), Inter-query Parallelism, Distributed Join Algorithms (Semi-join, Hash Join), Query Shipping vs Data Shipping.

**Parallel and Distributed Transaction Processing:** Distributed ACID Transactions, Two-Phase Commit Protocol (2PC), Three-Phase Commit, Distributed Deadlock Handling, Global Serializability.

**Security:** Authentication and Authorization, Discretionary Access Control (DAC), Role-Based Access Control (RBAC), Mandatory Access Control (MAC), SQL Injection and Prevention, Encryption (TDE, column-level encryption), Auditing and Compliance.

Blockchain Databases, Temporal Databases, Logic Based Databases, Object-Based Databases XML, Information Retrieval.

### **Reading List:**

1) M. Tamer Özsu, Principles of Distributed Database Systems, Prentice Hall, 1999.

- 2) Ceri S and Pelagatti G, Distributed databases: Principles and Systems, McGraw Hill, 2000.
- 3) Thomas Connolly and Carolyn Begg, Database Systems: A Practical Approach to Design, Implementation and Management, Pearson Education, 2002.
- 4) NIT Surathkal <https://infotech.nitk.ac.in/course/it461>
- 5) Advanced databases NIT Warangal  
[https://www.nitw.ac.in/CSEDeptAssets/docs/07\\_btech-in-cse.pdf](https://www.nitw.ac.in/CSEDeptAssets/docs/07_btech-in-cse.pdf)
- 6) Advanced database systems NIT Mizoram  
[https://www.nitmz.ac.in/Computer%20Science%20and%20Engineering/CSE\\_COURSE\\_STRUCTURE\\_AND\\_SYLLABUS\\_DUAL\\_DEGREE.pdf](https://www.nitmz.ac.in/Computer%20Science%20and%20Engineering/CSE_COURSE_STRUCTURE_AND_SYLLABUS_DUAL_DEGREE.pdf)

<b>CS3431</b>	<b>Graph Algorithms</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-------------------------	------------	------------------	------------------

Students of this course upon completion will learn how to model problems as graph-based problems and to solve the problem. Students will also learn standard graph algorithms and notions in this course across a spectrum of problems from the conventional spanning trees, network flows to the contemporary social network analytics.

**Pre-requisites:**

- i. Design and Analysis of Algorithms (CS2031)
- ii. Discrete Mathematics (MA1051)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Formulate a given problem as a suitable graph-theoretic problem. (Analyze)
<b>CO2</b>	Construct efficient algorithms for standard graph-theoretic problems with an analysis of their time and space requirements. (Analyze)
<b>CO3</b>	Construct an algorithm to solve a given graph-theoretic problem by applying a standard graph theoretic algorithm in isolation or by a combining multiple standard algorithms in a pipeline. (Analyze)
<b>CO4</b>	Analyze the properties of a given social network using standard graph-theoretic notions and algorithms. (Analyze)
<b>CO5</b>	Construct and analyze algorithms for solving network-flow-based problems with a comprehension of the relationship between cuts and flows. (Analyze)

**Course Articulation Matrix:**

PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	S	M										S	S	S
<b>CO2</b>	S	S	M										S	S	S
<b>CO3</b>	S	S	M										S	S	S
<b>CO4</b>	S	S	M										S	S	S
<b>CO5</b>	S	S	M										S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

Basic definitions and notions in Graph theory, Representation of graphs, Motivational examples for graphs: Web search engines and page rank algorithm, Web crawlers and social networks.

Strongly Connected Components, Biconnected components, Tarjan's algorithm, Invasion percolation with a case study on simulating river networks, Prim-Dijkstra-Jarnik algorithm, Boruvka's algorithm. Johnson's algorithm, Suurballe's Algorithm, Case studies on Map path planning, Landmark-based distance estimation, Widest path problems.

Eulerian graphs and standard results relating to characterization. Hamiltonian graphs, Dirac theorem, Chavathal theorem, closure of graph. Non-Hamiltonian graph with maximum number of edges. Self-centered graphs and related theorems.

**Graph Coloring:** Chromatic number, Greedy heuristic for Graph coloring, Interval graphs, Perfect graphs, Chordal graphs.

Planar graphs, Euler's formula, maximum number of edges in a planar graph. Five colour theorems. Graph Isomorphism.

Travelling Salesperson Problem, MST-doubling heuristic for finding TSP, Christofide's algorithm, Dynamic programming formulation for TSP – a review.

Matching, Bipartite graphs, Hopcroft-Karp algorithm, Finding Vertex covers and Independent Sets using matching, Stable matching, Gale-Shapley algorithm for Stable matching.

Max flow-Min Cut theorem, Ford-Fulkerson's algorithm, Case Study: Bipartite maximum matching as a flow problem.

**Social network analytics:** Erdős numbers, the Oracle of Bacon, and the Milgram small-world experiment, Properties – sparsity and power law, centrality, degeneracy and k-cores, Network models - Erdős-Rényi model, ERGM, random graphs with fixed degrees, Barabási-Albert model, Kleinberg's model, Network Cliques, Bron-Kerbosch algorithm.

Case-studies from competitive programming venues.

**Reading List:**

1. Douglas B. West, Introduction to Graph Theory, Second Edition, Pearson, 2000.
2. M. C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, First Edition, Academic Press, 1997.
3. R. J. Wilson and J. J. Watkins, Graphs: An Introductory Approach, First Edition, Wiley, 1990.

<b>CS3441</b>	<b>RANDOMIZED ALGORITHMS</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	------------------------------	------------	------------------	------------------

This course introduces the need for randomized algorithms and the different design paradigms for constructing randomized algorithms. The Course will also emphasize on analyzing the expected time complexity and the nature and bounds on the error probabilities of the randomized algorithms.

**Pre-requisites:**

- i. Probability, Statistics and Stochastic Processes (MA2031)
- ii. Design and Analysis of Algorithms (CS2031)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Construct and Analyze algorithms using a probabilistic framework. (Analyze)
<b>CO2</b>	Assess the computational hardness of a problem with a comprehension of the randomized complexity classes. (Analyze)
<b>CO3</b>	Construct Las Vegas algorithms for a given problem and compute the expected running time. (Analyze)
<b>CO4</b>	Construct Monte-Carlo algorithms for a given problem and compute the probability of getting an incorrect output. (Analyze)
<b>CO5</b>	Design and analyze solutions for complex problems using randomization design paradigms like Foiling the Adversary, Abundance of Witnesses, Fingerprinting, Random Sampling, Amplification and Random Rounding. (Analyze)

**Course Articulation Matrix:**

PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	S	M	M									S	S	S
<b>CO2</b>	S	S	M	M									S	S	S
<b>CO3</b>	S	S	M	M									S	S	S
<b>CO4</b>	S	S	M	M									S	S	S
<b>CO5</b>	S	S	M	M									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

**Introduction to Randomized Algorithms:** Review on Algebra, Number theory, Combinatorics and Probability theory, Probabilistic analysis of Algorithms, Randomized Complexity classes and hardness of problems, Randomness as a source of efficiency-designing a communication protocol.

**Models of Randomized Algorithms:** Classification-Las Vegas, Monte-Carlo (one-sided error, bounded-error and unbounded-error), Classification of Randomized Algorithms for Optimization problems.

**Design Paradigms:** Foiling the Adversary, Abundance of Witnesses, Fingerprinting, Random Sampling, Amplification, Random Rounding.

## **Representative Algorithms:**

**Foiling the Adversary:** Universal Hashing.

**Fingerprinting:** Communication protocols, Verification of Matrix Multiplication, Equivalence of Two polynomials.

**Success Amplification and Random Sampling:** Min-Cut, Satisfiability and repeated random sampling.

**Abundance of Witnesses:** Primality Testing.

**Optimization & random rounding:** Max-SAT review, hybrid sampling & rounding.

## **Derandomization Techniques.**

## **Reading List:**

1. Juraj Hromkovic– Design and Analysis of Randomized Algorithms, First edition, Springer, 2005.
2. Randomized Algorithms – Rajeev Motwani, Prabhakar Raghavan, Cambridge University Press, 1995.
3. Introduction to Algorithms –Charles E. Leiserson, Thomas H. Cormen, Ronald L. Rivest and Clifford Stein, third edition, PHI, 2010.

<b>CS3451</b>	<b>Computer Graphics</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Data Structures and Algorithms (CS1051)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	To execute line and circle drawing algorithms (Apply)
<b>CO2</b>	To perform clipping operations (Apply)
<b>CO3</b>	To perform 2D and 3D transformations (Apply)
<b>CO4</b>	To perform operations on curves and surfaces (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	L	L									S	S	S
<b>CO2</b>	S	M	L	L									S	S	S
<b>CO3</b>	S	M	L	L									S	S	S
<b>CO4</b>	S	M	L	L									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Overview:** Overview Of Graphics Systems: Introduction – Visualization; Graphics pipeline; Graphics hardware: Display devices, input devices; Raster Graphics.

**Lines:** Line and circle drawing algorithms; Windowing and 2D/3D clipping. Cohen and Sutherland line clipping, Cyrus Beck clipping method; 2D and 3D Geometrical Transformations: scaling, translation, rotation, reflection; Viewing Transformations: parallel and perspective projection;

**Curves and Surfaces:** cubic splines, Bezier curves, B-splines, Parametric surfaces. Surface of revolution Sweep surfaces, Fractal curves and surfaces; Hidden-line/surface removal methods; illuminations model; shading: Gouraud, Phong; Introduction to Ray-tracing;

**Animation:** Programming practices with standard graphics libraries like OpenGL.

### **Reading List:**

- 1) John F Hughes;Andries Van Dam;Morgan McGuire;David F Sklar;James D Foley;Steven K
- 2) Feiner;Kurt Akeley Computer Graphics: Principles and Practice, 3e, Pearson
- 3) Hearn D. and P. Baker, Computer Graphics, C Version, Prentice-Hall, ISBN 0-13-530924-7.
- 4) P. Shirley and S. Marschner, Computer Graphics, A.K.Peters (Cengage Learning)
- 5) Dave Shreiner; Graham Sellers; John Kessenich; Bill Licea-Kane, OpenGL Programming Guide 8e, Addison-Wesley

<b>CS3461</b>	<b>ADVANCED OPERATING SYSTEMS</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-----------------------------------	------------	------------------	------------------

This course introduces students to design issues and techniques in modern operating systems (OS) such as distributed OS, Database OS, Multi-processor OS and Mobile OS.

**Pre-requisites:**

- i. Operating Systems (CS2021)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Construct and evaluate solutions to address design issues in multi-processor OS with respect to its architecture, process management, synchronization, allocation and memory management. (Analyze)
<b>CO2</b>	Analyze and construct techniques and algorithms to solve problems arising while building distributed OS such as minimizing communication, deadlock detection and prevention, mutual exclusion, election, clock synchronization, scheduling, memory and file management. (Analyze)
<b>CO3</b>	Construct and analyze OS for database systems with a comprehension of the underlying transaction model, and theoretical and algorithmic aspects of concurrency control. (Analyze)
<b>CO4</b>	Construct and evaluate components of mobile OS with a comprehension of the device architecture, kernel structure, native programming interfaces and unique issues such as power management. (Analyze)

**Course Articulation Matrix:**

PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	S	S	M	M								S	S	
<b>CO2</b>	S	S	S	M	M								S	S	
<b>CO3</b>	S	S	S	M	M								S	S	
<b>CO4</b>	S	S	S	M	M								S	S	

S: Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

**Multiprocessor Operating Systems:** System Architectures, Structures of OS, OS design issues, Process synchronization, Process Scheduling and Allocation, Memory management.

**Distributed Operating Systems:** System Architectures, Design issues, Communication models, Clock synchronization, Mutual exclusion, Election algorithms, Distributed Deadlock detection.

Distributed scheduling, Distributed shared memory, Distributed File system, Multimedia file systems, File placement, Caching.

**Database Operating Systems:** Requirements of Database OS, Transaction process model, Synchronization primitives, Concurrency control: theoretical aspects and algorithms.

**Mobile Operating Systems:** ARM and Intel architectures, Mobile OS Architectures, Underlying OS - Kernel structure and native level programming, Runtime issues, Approaches to power management.

## **Reading List:**

1. Mukesh Singhal, Niranjana Shivaratri, Advanced Concepts in Operating Systems: Distributed, Database, and Multiprocessor Operating Systems, McGraw Hill, 2017.
2. Andrew S. Tanenbaum, Herbert Bos, Modern Operating Systems, 5th Edition, Pearson, 2022.
3. Wikipedia, Mobile Operating Systems, General Books LLC, 2010.

<b>CS3471</b>	<b>Design and Analysis of Parallel Algorithms</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---	------------	------------------	------------------

Students of this course upon completion will learn design and analysis principles for constructing scalable parallel algorithms across various Flynn’s taxonomy of parallel computers. The course deals with problems from both numerical and non-numerical natures such as linear algebraic problems, graph theoretic problems, discrete optimization search related problems, and dynamic programming algorithms.

**Pre-requisites:**

- i. Design and Analysis of Algorithms (CS2031)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Analyze the resource requirements of parallel algorithms with a comprehension of the parallel computer features and the underlying type of interconnection network. (Analyze)
CO2	Infer the most suitable parallel algorithm among a set of candidate algorithms to solve a given problem based on metrics such as cost, speed-up, efficiency and scalability. (Analyze)
CO3	Construct parallel algorithms for solving numerical problems from domains like Linear Algebra having representative problems such as matrix multiplication, transpose, and direct and iterative methods for solving systems of linear equations. (Apply)
CO4	Construct parallel algorithms for solving problems from non-numerical domains like graph theory, search in discrete optimization, and monadic and polyadic dynamic programming problems. (Apply)
CO5	Construct implementations of parallel algorithms on top of message-passing based parallel programming frameworks like Message Passing Interface. (Apply).

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M										S	M	M
CO2	S	S	M										S	M	M
CO3	S	M	L										S	M	M
CO4	S	M	L										S	M	M
CO5	S	M	L										S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Introduction:** The need for parallel computers, Models of computation (SISD, MISD, SIMD, MIMD), Analyzing algorithms, Expressing Algorithms.

**Selection and Merging:** Introduction – Selection and Merging, The problem and a lower bound, A Sequential algorithm, Desirable properties for parallel algorithms, Broadcasting a datum, Computing All Sums, An algorithm for parallel selection, A network for merging, Merging on CREW model, Merging on the EREW model.

**Sorting:** Introduction, A network for sorting, Sorting on a linear array, Sorting on the CRCW model, Sorting on CREW model, Sorting on the EREW model.

**Searching:** Introduction, Searching a sorted sequence, EREW, CREW, CRCW searching, Searching a random sequence, Searching on SM SIMD computers, Searching on a Tree, Searching on a Mesh.

**Performance Analysis:** Amdahl's Law, Gustafson Barsis's law- Karp Flatt metric- Iso efficiency metric.

**Numerical problems and implementation:** Matrix operations, Transposition, Mesh Transpose, Shuffle Transpose, EREW Transpose, Matrix by Matrix multiplication, Mesh multiplication, Cube multiplication, CRCW multiplication, Matrix by Vector multiplication, Linear Array multiplication, Tree multiplication, Convolution, Solving systems of linear equations (SIMD/MIMD), Finding roots of non-linear equations.

**Graph Theory:** connectivity matrix computation, finding connected components, computing all-pair shortest paths, minimum spanning trees, Traversals: Parallelizing breadth first and depth first searches.

**Optimization problems:** Job sequencing with deadlines, Knapsack problem.

**MPI primitives** with case-studies on implementation of parallel algorithms.

**Reading List:**

- 1) S.G. Akl, "The design and analysis of parallel algorithms", Prentice Hall of India, 1989.
- 2) Michael Jay Quinn, "Parallel programming in C with MPI and OpenMP", McGraw-Hill Higher Education, 2004.
- 3) S. Lakshmivarahan and S.K. Dhall, "Analysis and design of parallel algorithms – Arithmetic and Matrix problems", McGraw Hill, 1990.

<b>CS3481</b>	<b>Advanced Data Structures</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---------------------------------	------------	------------------	------------------

This course exposes a student to data structures that support persistence, temporality, dynamic optimality, cache-efficiency and for designing optimal geometric, integer, and string structures.

**Pre-requisites:**

- i. Data Structures and Algorithms (CS1051)
- ii. Design and Analysis of Algorithms (CS2031)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Construct and analyze persistent data structures with a comprehension of bounded-degree pointer machine model. (Analyze)
<b>CO2</b>	Construct and analyze retroactive temporal data structures that can support various retro-activity types such as full, partial and non-oblivious. (Analyze)
<b>CO3</b>	Construct and analyze suitable data structures for solving Computational Geometric problems. (Analyze)
<b>CO4</b>	Construct and analyze suitable data structures that are dynamically optimal. (Analyze)
<b>CO5</b>	Construct and analyze cache-oblivious and cache-efficient data structures. (Analyze)
<b>CO6</b>	Construct and analyze suitable data structures for representing strings and integers for efficient search and retrieval operations. (Analyze)
<b>CO7</b>	Construct and analyze dynamic graph structures. (Analyze)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M									S	S	
CO2	S	S	S	M									S	S	
CO3	S	S	S	M									S	S	
CO4	S	S	S	M									S	S	
CO5	S	S	S	M									S	S	
CO6	S	S	S	M									S	S	
CO7	S	S	S	M									S	S	

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Persistence:** pointer machine model, partial persistence, full persistence, confluent persistence, functional.

**Temporality:** Partial retroactivity, full retroactivity, non-oblivious retroactivity.

**Computational Geometry:** Point location via persistence, dynamic via retroactive; orthogonal range queries, range trees, layered range trees, dynamizing augmentation via weight balance, fractional cascading, Quadtrees, Kinetic data structures.

**Dynamic optimal Structures:** Binary search trees, analytic bounds, splay trees, geometric view, greedy algorithm, Independent rectangle, wilber, and signed greedy lower bounds; key-independent optimality; Tango trees.

**Cache-oblivious structures:** Models, cache-oblivious B-trees, Ordered-file maintenance, list labeling, order queries, cache-oblivious priority queues, Distribution sweeping via lazy funnel sort; cache-oblivious orthogonal 2D range searching: batched and online.

**Tree Structures for Sets of Intervals:** Interval Graphs, Interval Trees, Segment Trees-Trees for union of intervals, Trees for sums of weighted intervals, Trees for interval restricted Maximum sum queries, Orthogonal Range Trees-Higher Dimensional Segment Trees, Other systems of building blocks, Range counting and the semi group model.

**Integer Structures:** Hashing: Perfect Hashing, Universal Hashing, Cuckoo Hashing; Models, predecessor problem, van Emde Boas, x-fast and y-fast trees, indirection, Fusion trees: sketching, parallel comparison, Predecessor lower bound via round elimination, Sorting in linear time for  $w = O(\lg^2 + \epsilon n)$ -bit integers; Heaps: Heaps of optimal complexity, Double Ended Heap structures and Multi-dimensional Heaps, Heap related structures with Constant-Time updates.

**Data Structures for Strings:** Suffix Trees, Suffix Arrays, Linear Time construction of Suffix Trees, Aho-Corasick algorithm, Compact string structures.

**Dynamic Graphs:** Link-cut trees, heavy-light decomposition, Euler tour trees, decremental connectivity in trees in  $O(1)$ , fully dynamic connectivity in  $O(\lg^2 n)$ , survey,  $\Omega(\lg n)$  lower bound for dynamic connectivity.

### **Reading List:**

- 1) Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, second Edition, Pearson, 2004.
- 2) Michael T. Goodrich, Roberto Tamassia, Algorithm Design, First Edition, Wiley, 2006.
- 3) Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Second Edition, PHI, 2009.
- 4) Peter Brass, Advanced Data Structures, First Edition, Cambridge University Press, 2008.
- 5) Reinhard Kuzelnigg, Random Graphs and Cuckoo Hashing, First Edition, Sudwestdeutscher Verlag Fur Hochschulschriften AG, 2009.
- 6) Research papers.

<b>CS3491</b>	<b>Cloud Computing</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Operating Systems (CS2021)
- ii. Software Development (CS2061)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Determine Cloud Computing Architectures and Services for various societal applications based on financial and technological implications (Apply)
<b>CO2</b>	Analyze current state of the art Cloud infrastructures. (Analyze)
<b>CO3</b>	Develop private and hybrid cloud for organizations to execute customized applications (Apply)
<b>CO4</b>	Analyze authentication, confidentiality and privacy issues in Cloud computing environment. (Analyze)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	L	L	L								S	S	S
<b>CO2</b>	S	M	L	L	L								S	S	S
<b>CO3</b>	S	M	L	L	L								S	S	S
<b>CO4</b>	S	M	L	L	L								S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Introduction** - Cloud Computing Architecture, Cloud Delivery Models, The SPI Framework, SPI Evolution, The SPI Framework vs. the Traditional IT Model, Cloud Software as a Service (SaaS), Cloud Platform as a Service (PaaS), Cloud Infrastructure as a Service (IaaS) Google Cloud Infrastructure - Google File System – Search engine – MapReduce.

**API Design:** Amazon Web Services - REST APIs - SOAP API - Defining Service Oriented Architecture, Combining the cloud and SOA, Characterizing SOA, Loosening Up on Coupling, Making SOA Happen, Catching the Enterprise Service Bus, Telling your registry from your repository, Cataloging services, Understanding Services in the Cloud. Serving the Business with SOA and Cloud Computing, Query API.

**Connection:** User Authentication- Connecting to the Cloud - OpenSSH Keys - Tunneling / Port Forwarding.

**Storage:** Simple Storage Service - S3, EC2 - EC2 Compute Units, Platforms and storage, EC2 pricing, EC2 customers Amazon Elastic Block Storage - EBS - Ubuntu in the Cloud - Apache Instances in EC2 – Amazon Cloud Services- Amazon Elastic Compute Cloud (Amazon EC2), Amazon SimpleDB, Amazon Simple Storage Service (Amazon S3), Amazon CloudFront, Amazon Simple Queue Service (Amazon SQS), Amazon Elastic MapReduce, Amazon Relational Database Service (Amazon RDS) , EC2 Applications – Web application design - AWS EC2 Capacity Planning – Apache Servers - Mysql Servers – Amazon Cloud Watch - Monitoring Tools.

**Reading List:**

- 1) Anothony T Velte, Toby J Velte, Robert Elsenpeter, Cloud Computing: A Practical Approach, MGH, 2010.
- 2) Arshdeep Bahga, Vijay Madisetti, Cloud Computing: A Hands-on Approach, Universities Press (India) Private Limited, 2014.
- 3) Gautam Shroff, Enterprise Cloud Computing, Cambridge, 2010.
- 4) Rajkumar Buyya, James Broberg, Andrzej Goscinski, Cloud Computing Principles and Paradigms, Wiley, 2011.
- 5) Ronald Krutz and Russell Dean Vines, Cloud Security, 1st Edition, Wiley, 2010.

<b>CS3501</b>	<b>Model-driven Frameworks</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Software Development (CS2061)
- ii. Software Engineering (CS3021)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Construct domain specific languages (Apply)
<b>CO2</b>	Construct model transformations and synthesize model metrics (Apply)
<b>CO3</b>	Understand contemporary approaches to model driven engineering (Understand)
<b>CO4</b>	Apply domain specific modeling approach to authentic cases (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	M	M	L	L									S	S	S
<b>CO2</b>	M	M	L	L									S	S	S
<b>CO3</b>	M	M	L	L									S	S	S
<b>CO4</b>	M	M	L	L									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Traditional software engineering approach:** Drawbacks, Software processes, modular-based software design.

**Model Driven Software Engineering (MDSE) Principles:** MDSE basis, Overview of MDSE Technology, Criticism of MDSE, MDSE use cases.

**Model driven Architecture (MDA):** MDA Definitions and Assumptions, the modeling levels-CIM, PIM, PSM, mapping, general purpose and domain specific language in MDA, architecture Driven modernization.

**Integration of MDSE in a development process:** introducing MDSE in a software development process, traditional development process and MDSE, Domain driven design and MDSE, Test driven Development and MDSE.

**Modeling Language at a glance:** Anatomy of modeling language, general purpose vs. domain specific modeling language, General purpose modeling- the case of UML, UML profile platforms, software artifacts using UML standard modeling language, defining modelling constraints, automated GUI generation.

**Transformations:** Model to model transformations, model to text transformations.

**MDA Practice, Usage of QVT, Kermeta, etc.,** MDA Transformation Languages, model editors, model valuator, model metrics, modeling framework, middleware to support transformations, MDA applications.

**Reading List:**

- 1) Thomas Stahl, Markus Voelter, Model-Driven Software Development: Technology, Engineering, Management, Wiley, 2006.
- 2) Kleppe, Jos Warmer, and Wim Bast, MDA Explained - The Model Driven Architecture: Practice and Promise, Pearson Education, Boston, USA, 2003.

<b>CS3511</b>	<b>Cyber Laws and IPR</b>	<b>DEC</b>	<b>3-0-0</b>	<b>3 Credits</b>
---------------	---------------------------	------------	--------------	------------------

**Pre-requisites:** None

**Course Outcomes:** At the end of the course, student will be able to:

CO1	Assess the cyber laws in general and Indian IT act in particular. (Analyze)
CO2	cybercrimes and their culpability under various sections of the act. (Analyze)
CO3	Examine cyber case laws and recall various cases for developing solutions. (Analyze)
CO4	Assess the intellectual property rights in Indian context. (Analyze)
CO5	Extract illegal knowledge related to computer-based activities and diffuse such knowledge. (Apply)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M				L					S	S	S
CO2	S	S	S	M				L					S	S	S
CO3	S	S	S	M				L					S	S	S
CO4	S	S	S	M				L					S	S	S
CO5	S	M	L					L					S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

Cyber Space- Fundamental definitions -Interface of Technology and Law.

Jurisprudence and-Jurisdiction in Cyber Space - Indian Context of Jurisdiction - Enforcement agencies – Need for IT act - UNCITRAL – E-Commerce basics.

Information Technology Act, 2000 - Aims and Objects — Overview of the Act – Jurisdiction - Electronic Governance – Legal Recognition of Electronic Records and Electronic Evidence - Digital Signature Certificates - Securing Electronic records and secure digital signatures - Duties of Subscribers - Role of Certifying Authorities -

Regulators under the Act -The Cyber Regulations Appellate Tribunal - Internet Service Providers and their Liability – Powers of Police under the Act – Impact of the Act on other Laws.

**Cyber Crimes** -Meaning of Cyber Crimes –Different Kinds of Cybercrimes – Cybercrimes under IPC, CRPC and Indian Evidence Law - Cybercrimes under the Information Technology Act, 2000 - Cybercrimes under International Law - Cyber Stalking, Virus Dissemination,

Software Piracy, Internet Relay Chat (IRC) Crime, Credit Card Fraud, Net Extortion - Cyber Terrorism - Violation of Privacy on Internet - Data Protection and Privacy – Indian Court Case Studies.

### **Intellectual Property Rights**

Copyrights- Software – Copyrights vs Patents debate - Authorship and Assignment Issues - Copyright in Internet - Multimedia and Copyright issues - Software Piracy –

Trademarks - Trademarks in Internet – Copyright and Trademark cases

Patents - Understanding Patents - European Position on Computer related Patents - Legal position on Computer related Patents - Indian Position on Patents – Case Law

Domain names - Registration - Domain Name Disputes-Cyber Squatting-IPR cases

### **Reading List:**

- 1) Justice Yatindra Singh: Cyber Laws, Universal Law Publishing Co., New Delhi.
- 2) Farouq Ahmed, Cyber Law in India, New Era publications, New Delhi.
- 3) S.R.Myneni: Information Technology Law (Cyber Laws), Asia Law House, Hyderabad.
- 4) Chris Reed, Internet Law-Text and Materials, Cambridge University Press.
- 5) Pawan Duggal: Cyber Law- the Indian perspective Universal Law Publishing Co., New Delhi.

III Year II Semester Department Electives

<b>CS3701</b>	<b>Full Stack Development</b>	<b>DEC</b>	<b>3 - 0 - 0</b>	<b>3 Credits</b>
---------------	-------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Software Development (CS2061)
- ii. Software Engineering (CS3021)

**Course Outcomes: At the end of this course, students will be able to:**

CO1	Analyze the basic web components and development environment.
CO2	Construct a simple interactive web application using both client side and server-side scripting.
CO3	Build a simple web application using RESTful web service
CO4	Examine the need of security in developing web applications

**Detailed Syllabus:**

**Basics of Full Stack:** Understanding the Basic Web Development Framework, User, Browser, Webserver, Backend Services, MVC Architecture, Understanding the different stacks, The role of Express, Angular, Node, MongoDB, React.

**HTML and Java Script:** Introduction to HTML, Basic structure of an HTML document, creating an HTML document, Working with Lists, Tables and Frames, Working with Hyperlinks, Images and Multimedia, Working with Forms and controls, Introduction to JavaScript, Basic Syntax, Control Structures, Writing Functions, Working with Arrays, The Document Object Model, Events Handling, Client-side Validation, Form Validation & RegExps, ASP, Perl CGI, & Form Methods, SSI & Cookies, Frames & Windows, mimeTypeypes, plugins.

**Node Js:** Basics of Node JS, Installation, Working with Node packages, Using Node package manager, Creating a simple Node.js application, Using Events, Listeners, Timers, Callbacks, Handling Data I/O, Implementing HTTP services in Node.js.

**MongoDB:** Understanding NoSQL and MongoDB, Building MongoDB Environment, User accounts, Access control, Administering databases, Managing collections, Connecting to MongoDB from Node.js, simple applications.

**Express and Angular:** Implementing Express in Node.js, Configuring routes, Using Request and Response objects, Angular, Typescript, Angular Components, Expressions, Data binding, Built-in directives.

**React:** MERN STACK, Basic React applications, React Components, React State, Express REST APIs, Modularization and Webpack, Routing with React Router, Server-side rendering.

**Web security model** - HTTPS: Goals and Pitfalls - Injection flaws: Cross site scripting, SQL Injection, Command Injection, Cross site request forgery and HTTP header injection - Cookie Flaws and Server Misconfiguration - Session management and user authentication.

### Reading List:

1. Deitel, Deitel and Nieto, "Internet and Worldwide Web - How to Program", 5th Edition, PHI, 2011.
2. Brad Dayley, Brendan Dayley, Caleb Dayley, 'Node.js, MongoDB and Angular Web Development', Addison-Wesley, Second Edition, 2018.
3. Vasan Subramanian, 'Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node', Second Edition, Apress, 2019.
4. Chris Northwood, 'The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer', Apress; 1st edition, 2018.
5. Kirupa Chinnathambi, 'Learning React: A Hands-On Guide to Building Web Applications Using React and Redux', Addison-Wesley Professional, 2nd edition, 2018.

<b>CS3711</b>	<b>Virtual Reality and Augmented Reality</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--	------------	------------------	------------------

**Pre-requisites:**

- i. Computer Graphics (CS3451)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Understand the basics of AR VR perception (Understand)
<b>CO2</b>	Integrate sensors with AR/VR system (Apply)
<b>CO3</b>	Design AR/VR application for a given task (Apply)
<b>CO4</b>	Analyse existing AR/VR systems (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	L	M	L	L									S	S	S
<b>CO2</b>	L	M	L	L									S	S	S
<b>CO3</b>	L	M	L	L									S	S	S
<b>CO4</b>	L	M	L	L									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Introduction:** Virtual Reality, History, Overview of Various Realities, Immersion.

**Perception:** Objective & Subjective Reality, Perceptual Modalities, Perception of Space & Time.  
Content Creation: Environmental design, Affecting Behavior, VR Content Creation.

**Interaction:** VR Interaction Concepts, Input Devices, Interaction Patterns & Techniques. Iterative Design of VR: Philosophy of Iterative Design, The Define stage, Make stage and Learn stage.

**Software Development in Virtual Reality:** Introduction to Augmented Reality, Displays, Tracking, Computer Vision for Augmented Reality, Calibration & Registration, Visualization, Interaction, Modeling & Annotation, Authoring, Navigation, Collaboration.

**Software Development in Augmented Reality:** Applications of VR/AR in Entertainment, Medical, Manufacturing, Education, etc., Future of VR/AR.

### **Reading List:**

- 1) Jason Jerald, "The VR Book: Human-Centered Design for Virtual Reality", ACM and Morgan & Claypool Publishers, 2015.
- 2) Dieter Schmalstieg, Tobias Hollerer, "Augmented Reality, Principles and Practice", Addison Wesley (Pearson Education), 2016.
- 3) Jesse Glover and Jonathan Linowes, "Complete Virtual Reality and Augmented Reality Development with Unity", Packt Publishers, 2019.

<b>CS3721</b>	<b>Software Testing</b>	<b>DEC</b>	<b>3-0-0</b>	<b>3 Credits</b>
---------------	-------------------------	------------	--------------	------------------

**Pre-requisites:**

- i. Software Engineering (CS3021)
- ii. Software Development (CS2061)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Analyze various test processes and continuous quality improvement. (Analyze)
<b>CO2</b>	Classify and analyze types of bugs, errors and faults based on severity levels. (Analyze)
<b>CO3</b>	Model the behaviour of the software application using FSM. (Apply)
<b>CO4</b>	Apply software testing techniques suitable to the environment used for building the software. (Apply)
<b>CO5</b>	Identify the suitable testing tools for an application to generate reports using that tool. (Analyze)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	M	M	L	M	L	L		L	M	M		L	M	M	
<b>CO2</b>	M	M	M	S	M	L		L	L	L		L	M	M	
<b>CO3</b>	S	L	M	S	M	L		L	M	L		L	M	M	
<b>CO4</b>	M	M	L	S	M	L		L	M	L		L	M	M	
<b>CO5</b>	M	M	L	M	M	L		L	L	M		L	M	M	

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Introduction:** Testing as an Engineering Activity, Principles of Testing, Vmodel concepts, Tester's Role in a Software Development Organization, Bugs, Origins of Defects, Cost of defects, Defect Classes, The Defect Repository and Test Developing a Defect Repository, Testing strategies and techniques: Design, Developer/Tester Support of Defect Prevention strategies.

**Testing Strategies and Techniques:** Unit Testing, integration testing, system and acceptance testing, performance testing, regression testing, internationalization testing, ad hoc testing, object-oriented testing, Usability and Accessibility Testing.

**Black Box Approach to Test Case Design:** Random Testing, Requirements based testing, Equivalence partitioning, boundary value analysis, cause effect graphing, state-based testing, domain testing.

**White Box Approach to Test design:** Test Adequacy Criteria, static testing vs. structural adequacy criteria based on control flow, principles of mutation testing, equivalent mutants, fault detection using mutation, Test assessment using mutation.

**Test Management:** People and organizational issues in testing, Organization structures for testing teams, testing services, Test Planning, Test Plan Components, Test Plan Attachments, Locating Test Items, test management, test process, Reporting Test Results, Introducing the test specialist Test Automation Skills needed by a test specialist, Building a Testing Group.

**Software Test Automation:** Software test automation, skill needed for automation, scope of automation, design and architecture for automation, requirements for a test tool, challenges in automation, Test metrics and measurements, project, progress and productivity metrics.

#### **Reading List:**

- 1) Aditya P. Mathur “Foundations of Software Testing”, Second Edition, Pearson 2014.
  - 2) Srinivasan Desikan, Gopalaswamy Ramesh, “Software testing Education, principles and practices”, First Edition, Pearson Education, 2009
- Naresh Chauhan, Software Testing, Principles and practices, Oxford Press, Second Edition, 2016.

CS3731	Wireless Technologies	DEC	3 – 0 – 0	3 Credits
--------	-----------------------	-----	-----------	-----------

**Pre-requisites:**

- i. Computer Networks (CS3031).

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Differentiate wired and wireless technologies when selecting suitable technology for given requirements. (Analyze)
CO2	Implement suitable Wireless Local loop Technology. (Apply)
CO3	Apply suitable techniques in the selection of CDMA or TDMA. (Apply)
CO4	Apply suitable techniques in the selection of suitable satellite parameters and configurations. (Apply)
CO5	Apply suitable IEEE standards when selecting wireless LAN technology. (Apply)
CO6	Analyze different IEEE Medium Access Control protocols and choose a suitable one for the requirements. (Analyze)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M		S				M				S	S	M
CO2	S	M	M		S				M				S	S	M
CO3	S	M	M		S				M				S	S	M
CO4	S	M	M		S				M				S	S	M
CO5	S	S	S	M	S				M				S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Technical Review- Fundamentals:** Transmission Fundamentals, Signals for Conveying Information, Analog and Digital Data Transmission, Channel Capacity, Transmission Media.

**Multiplexing Communication Networks:** LANs, MANs, WANs, Switching Techniques, Circuit Switching, Packet Switching.

**Asynchronous Transfer Mode Protocols and the TCP/IP Suit:** The Need for a Protocol Architecture, The TCP/IP Protocol Architecture, The OSI Model, Internetworking.

**Wireless Networking- Satellite Communications:** Satellite Parameters and Configurations.

**Capacity Allocation:** Frequency Division, Time Division.

**Cellular Wireless Networks:** Principles of Cellular Networks, First-Generation Analog, Second-Generation TDMA, Second-Generation CDMA, Third-Generation Systems.

**Cordless Systems and Wireless Local Loop:** Cordless Systems, Wireless Local Loop, WiMAX, and IEEE 802.16 Broadband Wireless Access Standards.

**Mobile IP and Wireless Access Protocol:** Mobile IP, Wireless Application Protocol.

**Wireless LANs- Wireless LAN Technology:** Overview, Infrared LANs, Spread Spectrum LANs, Narrowband Microwave LANs Wi-Fi and the IEEE 802.11.

**Wireless Lan Standard:** IEEE 802 Protocol Architecture, IEEE 802.11 Architecture and Services, IEEE 802.11 Medium Access Control, IEEE 802.11 Physical Layer, Other IEEE 802.11 Standards.

**Bluetooth and IEEE 802.15:** Overview, Radio Specification, Baseband Specification, Link Manager Protocol, Logical Link Control and Adaptation Protocol, IEEE 802.15.

### **Reading List:**

- 1) W. Stallings, "Wireless Communications and Networks," Second Edition, Pearson Education, 2002.
- 2) T S Rappaport, "Wireless Communications: Principles & Practice," Second Edition, Pearson Education, 2002.
- 3) J Schiller, "Mobile Communications," Second Edition, Addison Wesley, 2000.
- 4) V K Garg, "IS-95 CDMA and CDMA2000", First Edition, Prentice Hall PTR, 2007.

<b>CS3741</b>	<b>Social Network Analytics</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Machine Learning (CS3011)
- ii. Design and Analysis of Algorithms (CS2031)

**Course Outcomes:** At the end of the course the student will be able to:

<b>CO1</b>	Compute network measures for a social media dataset (Apply)
<b>CO2</b>	Evaluate various diffusion models and influence techniques (Apply)
<b>CO3</b>	Construct techniques for community detection. (Apply)
<b>CO4</b>	Construct network measures for recommender systems (Apply)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	M	M									S	S	S
<b>CO2</b>	S	M	M	M									S	S	S
<b>CO3</b>	S	M	M	M									S	S	S
<b>CO4</b>	S	M	M	M									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Introduction to Social Network Mining, Graph Models:** Graph Essentials, Degree and Degree Distribution, Graph Representations, Types of Graphs, Connectivity in Graphs, Graph/Tree Traversal, Shortest Path Algorithms, Minimum Spanning Trees, Network Flow Algorithms.

**Network Measures:** Centrality, Degree Centrality, Eigenvector Centrality, Katz Centrality, PageRank, Topic-Specific PageRank, Betweenness Centrality, Closeness Centrality, Group Centrality, Transitivity and Reciprocity, Balance and Status, Similarity, Structural Equivalence, Regular Equivalence.

**Network Models:** Properties of Real-World Networks, Degree Distribution, Clustering Coefficient, Average Path Length, Random Graphs, Modeling Real-World Networks with Random Graphs, Small-World model, Modeling Real-World Networks with the Small-World Model, Preferential Attachment Model, Modeling Real-World Networks with the Preferential Attachment Model.

**Communities and Interactions:** Community Detection, Community Detection Algorithms, Member-Based Community Detection, Group-Based Community Detection, Community Evolution, Community Detection in Evolving Networks.

**Information Diffusion in social media:**

Information Cascades, Independent Cascade Model (ICM), Maximizing the Spread of Cascades, Diffusion of Innovations, Modeling Diffusion of Innovations, Epidemics, SI Model, SIR Model, SIS Model, SIRS Model.

**Influence and Homophily:** Measuring Assortativity, Measuring Assortativity for Nominal Attributes, Measuring Assortativity for Ordinal Attributes, Influence, Homophily, Distinguishing Influence and Homophily.

**Recommendation in social media:**

Classical Recommendation Algorithms, Content-Based Methods, Collaborative Filtering (CF), Extending Individual Recommendation to Groups of Individuals, Recommendation Using Social Context, Extending Classical Methods with Social Context, Evaluating Recommendations, Evaluating Relevancy of Recommendations, Evaluating Ranking of Recommendations.

**Reading List:**

- 1) Reza Zefarani, Mohammad Ali Abbasi, Huan Liu, "Social Media Mining: An Introduction." Cambridge University Press, 2014. ISBN: 978-1107018853.
- 2) Jure Leskovec, Anand Rajaraman and Jeffrey David Ullman, Mining of massive datasets, Cambridge University Press, 2014

<b>CS3751</b>	<b>Information Retrieval</b>	<b>DEC</b>	<b>3-0-0</b>	<b>3 Credits</b>
---------------	------------------------------	------------	--------------	------------------

**Pre-requisites:**

- i. Machine Learning (CS3011)
- ii. Design and Analysis of Algorithms (CS2031)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Identify the efficient IR model for an application or its dataset. (Apply)
<b>CO2</b>	Apply text processing, indexing, clustering and classification techniques to a corpus of texts for a specific information need (Apply)
<b>CO3</b>	Apply the techniques for web searching. (Apply)
<b>CO4</b>	Evaluate how techniques of IR integrate with recommender system. (Evaluate)
<b>CO5</b>	Evaluate how techniques of IR integrate with Question Answering. (Evaluate)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	L	L									S	M	L
<b>CO2</b>	S	M	M	L									S	M	L
<b>CO3</b>	S	M	M	L									S	M	L
<b>CO4</b>	S	S	M	M									S	S	M
<b>CO5</b>	S	S	M	M									S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Introduction to IR and Text pre-processing:**

Introduction, Data vs Information Retrieval, Logical view of the documents, Architecture of IR System, Web search system, Tokenization, Text Normalization, Stop-word removal, Morphological Analysis, Word Stemming, Lemmatization, Index term selection, Inverted indices, Positional Inverted index, Basic NLP tasks – POS tagging; shallow parsing.

**IR Models:** Classes of Retrieval Model, Boolean model, Term weighting mechanism – TF, IDF, TF-IDF weighting, Cosine Similarity, Vector space model, Probabilistic models, Non-Overlapping Lists, Proximal Nodes Model.

**Evaluation of IR:**

Precision, Recall, F-Measure, MAP (Mean Average Precision), Discounted Cumulative Gain, Known-item Search Evaluation.

**Query Operations and Languages:**

Relevance feedback and pseudo relevance feedback, Query expansion (with a thesaurus or WordNet and correlation matrix), Spelling correction, Query languages.

**Web Search:**

Search engines, Spidering (Structure of a spider, Simple spidering algorithm, multithreaded spidering, Bot), Directed spidering, Crawlers (Basic crawler architecture), Link analysis (HITS, Page ranking), Query log analysis, Handling “invisible” Web – Snippet generation, Cross Language Information Retrieval.

**Text Categorization and Clustering:**

Categorization, Learning for Categorization, General learning issues, Learning algorithms: Bayesian (naïve), Decision tree, KNN, Rocchio), Clustering algorithms (Hierarchical clustering, k-means, k-medoid, Expectation maximization (EM), Text shingling) .

**Recommender System:**

Personalization, Collaborative filtering recommendation, Content-based recommendation.

**Question Answering:**

Information bottleneck, Information Extraction, Ambiguities in IE, Architecture of QA system, Question processing, Paragraph retrieval, Answer processing.

**Reading List:**

- 1) Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008. <http://nlp.stanford.edu/IR-book/information-retrieval-book.html>
- 2) Modern Information Retrieval. Baeza-Yates Ricardo and Berthier Ribeiro-Neto. 2nd edition, Addison-Wesley, 2011.

CS3761	Service-Oriented Architecture	DEC	3-0-0	3Credits
--------	-------------------------------	-----	-------	----------

**Pre-requisites:**

i. Software Development (CS2061)

**Course Outcomes:** At the end of the course the student will be able to:

CO1	Design software-oriented architectures. (Apply)
CO2	Design medium scale software project development using SOA principles. (Apply)
CO3	Develop SOA messages from business use cases. (Apply)
CO4	Design and implementation of modern SOA and SOA-specific methodologies, technologies and standards. (Apply)
CO5	Create composite services by applying composition style. (Apply)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	L	M	L	S	M	L							S	S	
CO2	L	M	L	S	M	L							S	S	
CO3	L	M	L	M	M	L							S	S	
CO4	L	M	L	L	M	L							S	S	
CO5	L	M	L	S	M	L							S	S	

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Introducing SOA:** Fundamental SOA, Common characteristics of contemporary SOA, common misperceptions about SOA, common tangible benefits of SOA, common pitfalls of adopting SOA

**The evolution of SOA:** An SOA timeline (from XML to Web services to SOA), The continuing evolution of SOA (Standards organizations and Contributing vendors), The roots of SOA (comparing SOA to Past architectures).

**Web Services and Primitive SOA:** The Web services framework, Services (as Web services); Service descriptions (with WSDL); Messaging (with SOAP).

**Web Services and Contemporary SOA (Part-I: Activity Management and Composition) –** Message exchange patterns; Service activity; Coordination; Atomic Transactions; Business activities; Orchestration; Choreography. **Web Services and Contemporary SOA (Part-II: advanced messaging, Metadata and Security):** Addressing; Reliable messaging; Correlation; Policies; Metadata exchange; Security; Notification and eventing.

**Principles of Service - Orientation:** Services orientation and the enterprise; Anatomy of a service-oriented architecture; Common Principles of Service orientation; how service orientation principles interrelate; Service orientation and object orientation; Native Web service support for service orientation principles.

**Service Layers:** Service orientation and contemporary SOA; Service layer abstraction; Application service layer, Business service layer, Orchestration service layer; Agnostic services; Service layer configuration scenarios.

**An overview of JEE Web Services:** The JEE platform, The Technologies of Web Services, The JEE Web Service APIs.

XML Basics Primer Namespaces, Wrapping Up, The W3C XML Schema Language:XML Schema Basics, Advanced XML Schema.

XML based RPC, JSON, UDDI

**SOAP and WSDL:** The Basic Structure of SOAP, SOAP Namespaces, SOAP Headers, SOAP Body, SOAP Message Models, SOAP Faults, SOAP over HTTP.

### **Reading List:**

1) Thomas Erl, “Service-Oriented Architecture: Concepts, Technology and Design”, Prentice Hall Publication, 2005.

- 2) Richard Monson, Haefel, "J2EE Web Services", Pearson Education, 2004.
- 3) The Java EE 6 Tutorial: <https://docs.oracle.com/javaee/6/tutorial/doc/gijti.html>

<b>CS3771</b>	<b>PROBABILISTIC GRAPHICAL MODELS</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---------------------------------------	------------	------------------	------------------

Probabilistic graphical models (PGMs) are helpful in characterizing complex relationships among many random variates by fusing graph theory and probability theory and providing a sound framework. PGMs are helpful in reasoning under uncertainty especially in application domains like Natural Language Processing, Computer Vision, and Computational Biology.

**Pre-requisites:**

- i. Probability, Statistics and Stochastic Processes (MA2031)
- ii. Differential and Integral Calculus (MA1011)
- iii. Modelling and Optimization Techniques (EC2511)
- iv. Machine Learning (CS3011)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Construct probabilistic graphical models to characterize uncertainty inherent in problem domains such as Natural Language Processing and Computer Vision. (Apply)
<b>CO2</b>	Construct fundamental algorithms for probabilistic inference in both directed (Bayesian) and Undirected (Markovian) probabilistic graphical models. (Apply)
<b>CO3</b>	Construct algorithms for learning the structure and parameters of probabilistic graphical models. (Apply)
<b>CO4</b>	Apply Bayesian principles to model domain knowledge under uncertainty. (Apply)
<b>CO5</b>	Construct algorithms to perform inference in statistical and causal models. (Apply)

**Course Articulation Matrix:**

PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	M	L									S	M	S
<b>CO2</b>	S	M	M	L									S	M	S
<b>CO3</b>	S	M	M	L									S	M	S
<b>CO4</b>	S	M	M	L									S	M	S
<b>CO5</b>	S	M	M	L									S	M	S

S: Strong correlation, M: Medium correlation, L: Low correlation

## Detailed Syllabus:

**Introduction and Motivation:** Probabilistic Graphical Models, Representation, Inference and Learning.

**Representation:** Bayesian Networks: independence properties, D-separation Algorithm, I-equivalence, Minimal I-maps, Perfect maps.

**Undirected graphical models:** Parameterization, Gibbs distribution and Markov networks, Independencies in Markov networks, Conversion to/from Bayesian Networks to Markov Networks, partially directed models, Conditional Random Fields.

**Inference:** Exact inference-variable elimination, Clique Trees, Message passing, Belief update, Clique tree construction.

**Inference as Optimization:** propagation-based approximation.

**Particle-based approximate inference:** forward sampling, likelihood weighting and importance sampling, Markov Chain Monte Carlo methods, Gibbs Sampling algorithm, MAP Inference, Inference in hybrid networks and temporal models.

**Learning:** Learning graphical models: goals, learning as optimization.

**Parameter estimation:** Maximum Likelihood Estimation (MLE), MLE for Bayesian networks.

**Structure Learning in Bayesian Networks:** constraint-based approaches, structure search, scoring structures.

## Reading List:

- 1) Daphne Koller and Nir Friedman, Probabilistic Graphical Models, First Edition, MIT Press 2009.
- 2) Chris Bishop, Pattern Recognition and Machine Learning, First Edition, Springer 2006.

<b>CS3781</b>	<b>PERFORMANCE MODELING OF COMPUTER SYSTEMS</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---	------------	------------------	------------------

This course introduces the techniques of analytical modeling such as Markov chains, and queuing models, experimental designs, and discrete event simulations for modeling a computing system or its component and to analyze its performance with respect to well-defined metrics.

**Pre-requisites:**

- i. Discrete Event Simulation (CS2011)
- ii. Probability, Statistics and Stochastic Processes (MA2031)
- iii. Operating Systems (CS2021)
- iv. Differential and Integral Calculus (MA1011)
- v. Computer Organization and Architecture (CS1021)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Formulate performance models for a given computer and a communication system by applying modeling techniques like Markov Chains, Queuing theory and Queue networks. (Apply)
CO2	Design experiments to measure the performance of a computer system with an understanding of the appropriate performance metrics to be used. (Apply)
CO3	Model, characterize and reproduce workloads to a computer system. (Apply)
CO4	Construct discrete event simulation models for a computer system and analyze the simulation results. (Analyze)
CO5	Analyze, present, and interpret the experimental results to evaluate alternative system implementations. (Analyze)

**Course Articulation Matrix**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M										S	S	M
CO2	S	M	M										S	S	M
CO3	S	M	M										S	S	M
CO4	S	S	M										S	S	M

CO5	S	S	M										S	S	M
-----	---	---	---	--	--	--	--	--	--	--	--	--	---	---	---

S: Strong correlation, M: Medium correlation, L: Low correlation

### Detailed Syllabus:

**Overview of Performance Modeling:** introduction, Avoiding common mistakes, Selection of techniques and metrics.

**Modeling Techniques:** Random variables and probability distributions, Stochastic processes, Markov Chain, Continuous time Markov Chain, Queuing Theory, Queuing Networks.

**Workloads:** Types, Selection, Characterization Techniques-Averaging, Histograms, Clustering, Principal Component Analysis, Markov Models.

**Data Presentation:** Guidelines, Common mistakes, Pictorial Games, Gantt Charts, Kiviat Graphs-Shapes & Applications, Schumacher Charts.

**Experimental Design and Analysis:** 2k factorial designs, 2kr Factorial Designs with replications, 2kp Fractional Factorial designs.

**Simulation:** Analysis of simulation results – Model Verification Techniques, Validation, Stopping Criteria and Variance reduction, Discrete Event Simulations-Programming aspects.

**Queuing Models:** Queuing Theory, Analysis of a single queue, M/M/1 queue, M/M/m queue, M/M/m/B queue, Queuing Network models of computer systems, Operational laws, Analysis of queuing systems.

### Reading List:

1) Raj Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling", Wiley, 2015.

2) Mor Harchol-Balter, Performance Modeling and Design of Computer Systems: Queueing Theory in Action, Cambridge University Press, 2013.

CS3791	Computational Number Theory	DEC	3-0-0	3 Credits
--------	-----------------------------	-----	-------	-----------

**Pre-requisites:**

i. Discrete Mathematics (MA1051)

**Course Outcomes:** At the end of the course, student will be able to:

CO1	Analyse large integer computations in $Z_n$
CO2	Analyse primality testing and integer factorization algorithms
CO3	Develop algorithms for computations in groups, rings and fields
CO4	Develop algorithms for computations in polynomial rings
CO5	Develop algorithms for computations in finite fields

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	S	M	M	L	S	-	L	-	-	L	-	-	-	L	-	-
CO2	M	S	M	M	L	-	-	L	-	L	-	L	-	-	M	-
CO3	M	L	S	M	M	L	L	-	M	-	L	L	-	L	-	L
CO4	M	M	M	L	M	L	L	-	L	L	-	-	M	-	-	L
CO5	M	S	S	S	S	L	-	L	L	-	L	L	M	-	-	L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus**

**Algorithms for integer arithmetic:** Divisibility, gcd, modular arithmetic, modular exponentiation, Montgomery arithmetic, congruence, Chinese remainder theorem, Hensel lifting, orders and primitive roots, quadratic residues, integer and modular square roots, prime number theorem, continued fractions and rational approximations.

**Representation of finite fields:** Prime and extension fields, representation of extension fields, polynomial basis, primitive elements, normal basis, optimal normal basis, irreducible polynomials.

**Algorithms for polynomials:** Root-finding and factorization, Lenstra-Lenstra-Lovasz algorithm, polynomials over finite fields.

**Elliptic curves:** The elliptic curve group, elliptic curves over finite fields, Schoof's point counting algorithm.

**Primality testing algorithms:** Fermat test, Miller-Rabin test, Solovay-Strassen test, AKS test.

Finding generators and discrete logarithms - Finding a generator for  $Z_p$ , Computing discrete logarithms  $Z_p$ , The Diffie–Hellman key establishment protocol.

Quadratic residues and quadratic reciprocity - Quadratic residues, Legendre symbol, Jacobi symbol; Computational problems related to quadratic residues - Computing the Jacobi symbol, Testing quadratic residuosity, Computing modular square roots, The quadratic residuosity assumption.

**Integer factoring algorithms:** Trial division, Pollard rho method, p-1 method, CFRAC method, quadratic sieve method, elliptic curve method.

**Computing discrete logarithms over finite fields:** Baby-step-giant-step method, Pollard rho method, Pohlig-Hellman method, index calculus methods, linear sieve method, Coppersmith's algorithm.

**Applications:** Algebraic coding theory, cryptography.

### Reading List:

- 1) Victor Shoup, A Computational Introduction to Number Theory and Algebra, Cambridge University Press, 2008
- 2) Henri Cohen, A Course in Computational Algebraic Number Theory, Springer-Verlag, 2000
- 3) Abhijit Das, Computational Number Theory, Cambridge University Press, 2013
- 4) Eric Bach and Jeffrey Shallit, Algorithmic Number Theory, Volume 1: Efficient Algorithms, MIT Press, 1996
- 5) M. Mignotte, Mathematics for computer algebra, Springer-Verlag.
- 6) I. Niven, H. S. Zuckerman and H. L. Montgomery, An introduction to the theory of numbers, John Wiley.
- 7) J. von zur Gathen and J. Gerhard, Modern computer algebra, Cambridge University Press.
- 8) R. Lidl and H. Niederreiter, Introduction to finite fields and their applications, Cambridge University Press.
- 9) A. J. Menezes, editor, Applications of finite fields, Kluwer Academic Publishers.
- 10) J. H. Silverman and J. Tate, Rational points on elliptic curves, Springer International Edition.
- 11) D. R. Hankerson, A. J. Menezes and S. A. Vanstone, Guide to elliptic curve cryptography, Springer-Verlag.
- 12) A. Das and C. E. Veni Madhavan, Public-key cryptography: Theory and practice, Pearson Education Asia.

<b>CS3801</b>	<b>Design Patterns</b>	<b>DEC</b>	<b>3-0-0</b>	<b>3 Credits</b>
---------------	------------------------	------------	--------------	------------------

**Pre-requisites:**

- i. Software Development (CS2061)
- ii. Software Engineering (CS3021)

**Course Outcomes:** At the end of this course, students will be able to:

CO1	Analyze common design patterns in the context of incremental/iterative development (Analyze)
CO2	Evaluate software source code using patterns (Evaluate)
CO3	Analyze software design practices through design patterns (Analyze)
CO4	Implement the design patterns in an object-oriented language (Apply)
CO5	Apply various design pattern strategies to design an appropriate solution for the problem (Apply)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	S	M	S						L		S	M	M
CO2	S	M	S	M	S						L		S	M	M
CO3	S	M	S	M	S						L		S	M	M
CO4	S	M	S	M	S				M		L		S	M	M
CO5	S	M	S	M	S				M		L		S	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

Design Pattern, Architectural Design Patterns, Describing Design Patterns, The Catalog of Design patterns, selecting a Design Pattern, Using Design Patterns for Design problems

**Architectural Patterns** -- Layered architecture, Pipers and Filters, Blackboard, Broker, MVC, MVVM, Micro-Kernel, Master-Slave, PAC, others.

**Creational Patterns:** Singleton, Abstract Factory, Builder, Factory Method, Prototype, Implementation in various languages like Python, Java.

**Structural Pattern Part – I:** Adaptor, Bridge, and Composite.

**Structural Pattern Part – II:** Decorator, Arcade, Flyweight, Proxy.

**Behaviour Patterns Part – I:** Chain of Responsibility, Command, Interpreter, and Iterator.

**Behaviour Patterns Part – II:** Mediator, Memento, Observer.

**Behaviour Patterns Part – III** State, Strategy, Template Method, Visitor.

**Compound patterns, Case study:** using design patterns to solve an industry level problem.

#### **Reading List:**

- 1) Eric Freeman, Elisabeth Freeman, Kathy Sierra, Bert Bates “Head First Design Patterns”, O'Reilly, 2020.
- 2) Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides “Design Patterns:
- 3) Elements of Reusable Objectoriented Software” Addison-Wesley.
- 4) Mark Grand, “Patterns in JAVA Vol-I”, Wiley Dream Tech.
- 5) Mark Grand, “Patterns in JAVA Vol – II”, Wiley Dream Tech.
- 6) Mark Grand, “JAVA Enterprise Design Patterns Vol – III”, Wiley Dream Tech.
- 7) Rahul Verma, Chetan Giridhar, “Design patterns in Python”, Testing Perspective, 2011.
- 8) Martin Fowler, with Dave Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, and
- 9) Randy Stafford, “Patterns of Enterprise Application Architecture”, 1st Edition, Addison-Wesley

<b>CS3811</b>	<b>Advanced Algorithms</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	----------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Design and Analysis of Algorithms (CS2031)

This course exposes a student to advanced algorithmic design and analysis techniques involving network flows, approximate and fixed-parameter algorithms, computational geometry, cache-efficiency, LP, online, streaming, external and parallel algorithms.

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Construct and analyze algorithms for solving flow-based problems. (Analyze)
CO2	Formulate Linear Programming (LP) problems and apply simplex, interior-point and ellipsoid algorithms to solve LP problems. (Analyze)
CO3	Construct and analyze approximation and fixed-parameter algorithms for coping with NP-hardness. (Analyze)
CO4	Construct and analyze suitable online and streaming algorithm for solving a given problem. (Analyze)
CO5	Construct and analyze cache-oblivious and cache-efficient algorithms. (Analyze)
CO6	Construct and analyze algorithms for solving computational geometric problems. (Analyze)
CO7	Construct and analyze parallel algorithms and external memory based algorithms for dealing with sharded or big datasets. (Analyze)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M									S	S	
CO2	S	S	S	M									S	S	
CO3	S	S	S	M									S	S	
CO4	S	S	S	M									S	S	
CO5	S	S	S	M									S	S	
CO6	S	S	S	M									S	S	
CO7	S	S	S	M									S	S	

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Maximum Flows:**

Augmenting Paths and Push-Relabel Methods. Minimum Cost Flows. Bipartite Matching.

**Linear Programming:**

Formulation of Problems as Linear Programs. Duality. Simplex, Interior Point, and Ellipsoid Algorithms.

**Online Algorithms:**

Ski Rental. River Search Problem. Paging. The k-Server Problem. List Ordering and Move-to-Front.

**Approximation Algorithms:**

Coping with NP-Hardness. Greedy Approximation Algorithms. Dynamic Programming and Weakly Polynomial-Time Algorithms. Linear Programming Relaxations. Randomized Rounding. Vertex Cover, Wiring, and TSP.

**Fixed-Parameter Algorithms:**

Coping with NP-Hardness. Parameterized Complexity. Kernelization. Vertex Cover. Connections to Approximation.

### **Parallel Algorithms:**

PRAM. Pointer Jumping and Parallel Prefix. Tree Contraction. Divide and Conquer. Randomized Symmetry Breaking. Maximal Independent Set.

### **External-Memory Algorithms:**

Accounting for the Cost of Accessing Data from Slow Memory. Sorting. B-trees. Buffer Trees. Cache-oblivious Algorithms for Matrix Multiplication and Binary Search.

### **Computational Geometry:**

Convex Hull. Line-segment Intersection. Sweep Lines. Voronoi Diagrams. Range Trees. Seidel's Low-dimensional LP Algorithm.

### **Streaming Algorithms:**

Sketching. Distinct and Frequent Elements.

### **Reading List:**

- 1) Michael T. Goodrich, Roberto Tamassia, Algorithm Design, First Edition, Wiley, 2006.
- 2) Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein,
- 3) Introduction to Algorithms, Second Edition, PHI, 2009.
- 4) Ahuja, Magnanti, and Orlin. Network Flows. Upper Saddle River, NJ: Prentice Hall, 1993. ISBN: 013617549X.
- 5) Motwani and Raghavan. Randomized Algorithms. Cambridge, UK: Cambridge University Press, 1995. ISBN: 0521474655.
- 6) Gusfield, Dan. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge, UK: Cambridge University Press, 1997. ISBN: 0521585198.
- 7) Borodin, Allan, and Ran El-Yaniv. Online Computation and Competitive Analysis. Cambridge, UK: Cambridge University Press, 1998. ISBN: 0521563925.
- 8) Tarjan, Robert. Data Structures and Network Algorithms. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1983. ISBN: 0898711878. A classic - no longer up to date, but outstanding writing.
- 9) Berg, Mark de, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. Computational Geometry: Algorithms and Applications. New York, NY: Springer-Verlag, 2000. ISBN: 3540656200.
- 10) Hochbaum, Dorit, ed. Approximation Algorithms for NP-Hard Problems. Boston, MA: PWS Publishing Company, 1997. ISBN: 0534949681.
- 11) VV Vazirani. Approximation Algorithms: Springer-Verlag, 2001. ISBN: 3-540-65367-8.

<b>CS3821</b>	<b>REINFORCEMENT LEARNING</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-------------------------------	------------	------------------	------------------

This course introduces students to reinforcement learning techniques applicable in situations where trial and error attempts are required when explicit control signals are unavailable.

**Pre-requisites:**

- i. Machine Learning (CS3011)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Infer whether a given problem is amenable to be solved using RL as opposed to non-interactive vanilla ML and AI techniques. (Analyze)
CO2	Formulate a given problem as a RL problem with a specification of state space, action space, dynamics and reward model. (Analyze)
CO3	Infer which RL algorithm is best suited for addressing a given problem with justifications. (Analyze)
CO4	Implement and evaluate RL algorithms based on metrics such as regret, sample complexity, computational complexity, empirical performance, and convergence. (Analyze)
CO5	Infer suitable techniques for addressing the exploration vs. exploitation trade-off in terms of performance, scalability, complexity of implementation, and theoretical guarantees. (Analyze)

**Course Articulation Matrix**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M	M								S	S	S
CO2	S	S	S	M	M								S	S	S
CO3	S	S	S	M	M								S	S	S
CO4	S	S	S	M	M								S	S	S
CO5	S	S	S	M	M								S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

## Detailed Syllabus:

**Introduction to Reinforcement Learning:** Examples, Elements of RL, Limitations and Scope.

### Tabular Solution Methods:

**Multi-armed Bandits:** A k-armed bandit problem, Action-value methods, tracking a non-stationary problem, Optimistic initial values, Upper Confidence Bound Action selection, Gradient bandit algorithms, Contextual bandits.

**Finite Markov Decision Processes:** the agent-environment interface, goals and rewards, returns and episodes, Unified notation for episodic and continuous tasks, policies and value functions, optimality and approximation

**Dynamic Programming:** Policy evaluation, policy improvement, policy iteration, value iteration, asynchronous dynamic programming, generalized policy iteration, efficiency of dynamic programming.

**Monte Carlo methods:** Prediction, estimation of action values, Control and control without exploring starts, Off-policy prediction through importance sampling, incremental implementation, Off-policy MC control, Discounting-aware importance sampling, Per-decision importance sampling.

**Temporal Difference Learning:** prediction, advantages, optimality of TD(0), Sarsa: On-policy TD control, Q-learning: off-policy TD control, Expected Sarsa, Maximization bias and double learning, Games, after-states and special cases.

n-Step bootstrapping, Techniques for planning and learning with tabular methods

### Approximate-Solution methods:

On-policy prediction with approximation, On-policy control with approximation, Off-policy methods with approximation, Eligibility traces, Policy gradient methods.

**Applications and Case-studies:** TD-Gammon, Samuel's Checker Player, Watson's daily double wagering, Optimizing memory control, Human-level video game play, AlphaGo, Personalized web services, Thermal soaring.

Selected Research papers.

**Reading List:**

- 1) Sutton and Barto, Reinforcement Learning: An Introduction, 2nd Edition, MIT Press, 2018.
- 2) Marco Wiering and Martijn van Otterlo, Reinforcement Learning: State-of-the-Art, Springer, 2012.
- 3) Stuart J. Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, fourth edition, Pearson, 2022.

<b>CS3831</b>	<b>Deep Learning for Vision</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Machine Learning (CS3011)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Understand fundamental concepts of image processing and feature extraction techniques.
<b>CO2</b>	Application of deep learning techniques, including neural networks and optimization methods.
<b>CO3</b>	Convolutional neural networks (CNNs) are analyzed for classification, detection, and segmentation.
<b>CO4</b>	Evaluate attention-based models and transformers for vision applications
<b>CO5</b>	Development of generative and vision-language models for image synthesis and representation learning.

**Course Articulation Matrix:**

PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	S	M	L									S	L	L
<b>CO2</b>	S	M	M	L									S	M	L
<b>CO3</b>	S	S	M	L									S	M	L
<b>CO4</b>	S	S	M	L									S	M	L
<b>CO5</b>	S	S	S	M									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Introduction:** Introduction, Image Formation, Image Representation, Linear Filtering, Correlation, Convolution, Edge Detection, From Edges to Blobs and Corners, Scale Space, Image Pyramids and Filter Banks, SIFT and Variants, Human Visual System.

## Deep Learning Basics

**Neural Networks:** A Review, linear classifiers, Stochastic gradient descent, Feedforward Neural Networks and Backpropagation, Gradient Descent and Variants, Regularization in Neural Networks, Improving Training of Neural Networks.

## Convolutional Neural Networks

Introduction, Backpropagation in CNNs, CNN Architecture for Image Classification, Evolution of CNN Architectures: VGG, Inception, ResNets, ResNet Variants, MobileNet, EfficientNet, Finetuning CNNs, Visualizing CNNs, CNNs for Object Detection: Two-stage Models, CNNs for Object Detection: Single-stage Models, CNNs for Segmentation, Camera Calibration, Epipolar Geometry, Structure from Motion. Recurrent Neural Networks: Introduction.

## Attention Models and Transformers

**Attention in Vision Models:** An Introduction, Soft and Hard Attention: Image Captioning, Self-Attention and Transformers, From Transformers to Vision Transformers, Transformers for Detection, Transformers for Segmentation.

## Deep Generative Models

**Deep Generative Models:** An Introduction, Generative Adversarial Networks, GAN Hacks and Improvements, Variational Autoencoders and Disentanglement, Introduction to Diffusion Models: DDPMs, Classifier and Classifier-Free Diffusion Guidance, Text-conditioned Diffusion Models, Under the Hood: Sampling, Prediction Space, Noise Schedules, Architectures.

## Vision-Language Models and Recent Developments

**Self-Supervised Learning:** SimCLR, Contrastive Learning, Vision-Language Models, CLIP, BLIP, BLIP-2.

## Reading List:

- 1) Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 2016
- 2) Michael Nielsen, Neural Networks and Deep Learning, 2016
- 3) Yoshua Bengio, Learning Deep Architectures for AI, 2009
- 4) Richard Szeliski, Computer Vision: Algorithms and Applications, 2010.
- 5) Simon Prince, Computer Vision: Models, Learning, and Inference, 2012.
- 6) David Forsyth, Jean Ponce, Computer Vision: A Modern Approach, 2002.
- 7) Deep learning for Computer vision NPTEL: IIT Hyderabad
- 8) [https://onlinecourses.nptel.ac.in/noc24\\_cs89/preview](https://onlinecourses.nptel.ac.in/noc24_cs89/preview)

CS3841	Deep Learning for NLP	DEC	3 – 0 – 0	3 Credits
--------	-----------------------	-----	-----------	-----------

**Pre-requisites:**

- i. Machine Learning (CS3011)

**Course Outcomes:** At the end of the course the student will be able to:

CO1	To represent word meaning with word vectors such as Word2Vec and GloVe. (Apply)
CO2	To build neural language models (Apply)
CO3	To build seq-to-seq models for NLP applications (Apply)
CO4	To build Transformer based models for NLP applications (Apply)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	S	S	M	L							S	S	S
CO2	S	M	S	S	M	L							S	S	S
CO3	S	M	S	S	M	L							S	S	S
CO4	S	M	S	S	M	L							S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Introduction to NLP:** What is Natural Language Processing? A brief primer on word and sentence level tasks and n-gram language Model. Text classification using Naïve Bayes and Logistic regression, Multinomial Logistic Regression.

**Distributional semantics:** word2vec and Glove, embeddings visualization, semantic properties, bias and evaluation.

**Recurrent Neural Networks:** LSTM, GRU, Bidirectional and stacked variants, encoder decoder model with RNNs, Attention, Applications – sequence labelling and classification

**Transformers:** Self Attention, Transformer blocks, Parallelizing computations, Language modelling head.

**Large Language Models:** LLMs with Transformers, Sampling for LLM generation, Pretraining and evaluation of LLMs, Dealing with scale, Potential harms with LLMs.

**Masked Language Models:** BERT, T5, GPT, fine-tuning for classification, sequence labelling and NER.

**Model Alignment, Prompting and In-Context Learning:** Prompting, Model Alignment: Instruction Tuning, Chain-of-Thought Prompting, Automatic Prompt Optimization, Evaluation, RLHF and DPO for Alignment.

**NLP Applications:** Machine Translation, Question Answering, Chatbots and Dialog Systems etc.

### Reading List:

- 1) Daniel Jurafsky and James H. Martin. 2025. Speech and Language Processing. 3rd Edition.
- 2) <https://web.stanford.edu/~jurafsky/slp3/>
- 3) [https://pytorch.org/tutorials/beginner/deep\\_learning\\_nlp\\_tutorial.html](https://pytorch.org/tutorials/beginner/deep_learning_nlp_tutorial.html)
- 4) <https://huggingface.co/learn/nlp-course/>
- 5) Yoav Goldberg. [A Primer on Neural Network Models for Natural Language Processing](#)
- 6) Ian Goodfellow, Yoshua Bengio, and Aaron Courville. [Deep Learning](#)
- 7) Lewis Tunstall, Leandro von Werra, and Thomas Wolf. [Natural Language Processing with Transformers](#)

<b>CS3851</b>	<b>Generative AI</b>	<b>DEC</b>	<b>3-0-0</b>	<b>3 Credits</b>
---------------	----------------------	------------	--------------	------------------

**Pre-requisites:**

- i. Machine Learning (CS3011)

**Course Outcomes: At the end of this course, students will be able to:**

CO1	Develop a comprehensive understanding of the significance of generative model formulations.
CO2	Implement various generative models, including autoregressive models, VAEs, GANs, and Diffusion Models.
CO3	Design and implement generative models using widely-used frameworks.
CO4	Develop problem-solving skills by addressing challenges in the practical implementation of generative models.

**Detailed Syllabus:**

Introduction to Probabilistic Deep Generative Modelling, Autoregressive Models and Large Language Models Introduction, Maximum Likelihood Learning, Variational AutoEncoders, Latent Variable Models, Variational Inference, Normalizing Flow Models, Generative Adversarial Networks (GANs), Introduction to Diffusion Models, Energy based Models, Score based models, Diffusion beyond Denoising, Discrete Diffusion, Evaluation Metrics for Generative Models. Text-to-Image/Video Generation, Conditional Generation.

**Reading List:**

1. Foster D. Generative deep learning. " O'Reilly Media, Inc.; 2023
2. [Generative Modeling by Estimating Gradients of the Data Distribution](#) Yang Song. Blog post on score-based generative models, May 2021.
3. [How to Train Your Energy-Based Models.](#) Yang Song and Diederik P. Kingma. February 2021.
4. [Learning deep generative models.](#) Ruslan Salakhutdinov. Annual Review of Statistics and Its Application, April 2015.
5. [Stanford CS236](#) Lecture notes.
6. Y. Bengio, I. Goodfellow and A. Courville, "Deep Learning", MIT Press, 2016

7. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

**Resources used to prepare syllabus:**

1. <https://deepgenerativemodels.github.io/syllabus.html>
2. <https://cmu-dgm.github.io/schedule.html>
3. <https://user.ceng.metu.edu.tr/~gcinbis/courses/Spring24/CENG796/index.html>
4. <https://mit-6s978.github.io/schedule.html>
5. [https://study.iitm.ac.in/ds/course\\_pages/BSDA5002.html](https://study.iitm.ac.in/ds/course_pages/BSDA5002.html)
6. [https://github.com/davidADSP/Generative\\_Deep\\_Learning\\_2nd\\_Edition/tree/main/notebooks](https://github.com/davidADSP/Generative_Deep_Learning_2nd_Edition/tree/main/notebooks)

CS3861	Explainable AI	DEC	3 – 0 – 0	3 Credits
--------	----------------	-----	-----------	-----------

**Pre-requisites:**

- i. Machine Learning (CS3011)

**Course Outcomes:** At the end of the course the student will be able to:

CO1	To identify commonly used XAI techniques and their connection to Bias and Trustworthiness (Apply)
CO2	To develop and deploy model-agnostic perturbation methods and propagation-based feature importance methods (Apply)
CO3	To construct human-understandable explanations for model predictions. (Apply)
CO4	To design novel neural architectures that are interpretable by construction. (Apply)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	L	M	S	S									S	S	S
CO2	L	M	S	S									S	S	S
CO3	L	M	S	S									S	S	S
CO4	L	M	S	S									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Overview and taxonomy of XAI:** need for explainable and responsible AI, definition of terms, taxonomy of the XAI space.

Perturbation-based feature attribution methods such as LIME, SHAP and Anchors.

Propagation-based feature attribution methods such as Relevance Propagation and Saliency methods.

Concept-based explainability methods such as Net2Vec, T-CAV and ACE.

Interpretable architectures such as CBMs and variants, Concept Whitening and SENNs.

Neurosymbolic methods such as DeepProbLog and Neural Reasoners.

Sample-based Explanations such as Influence functions and ProtoPNets.

Counterfactual explanations.

### **Reading List:**

1) Christoph Molnar, "Interpretable Machine Learning". (2022):

2) <https://christophm.github.io/interpretable-ml-book/>

3) Arrieta, Alejandro Barredo, et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." Information fusion 58 (2020): 82-115.

4) Rudin, Cynthia, et al. "Interpretable machine learning: Fundamental principles and 10 grand challenges." Statistics Surveys 16 (2022): 1-85.

5) <https://courses.cs.washington.edu/courses/csep590b/22sp/files/lectures/lecture5.pdf>

6) <https://sites.google.com/view/aaai22-ximl-tutorial/home>

<b>CS3871</b>	<b>HIGH PERFORMANCE COMPUTING</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-----------------------------------	------------	------------------	------------------

This course introduces students to High-performance computing techniques to solve scientific, complex engineering and large-scale problems that are infeasible to be solved on a uniprocessor environment.

**Pre-requisites:**

i. Distributed Computing (CS3051)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Infer which HPC architecture is suitable for a given application or scientific workload, with a comprehension of the salient features of different HPC architectures. (Analyze)
CO2	Construct and evaluate the performance of parallel and distributed programs using techniques and platforms based on shared-memory and distributed-memory. (Analyze)
CO3	Analyze parallel and distributed HPC applications with an intent of identifying bottlenecks and enhancing scalability and efficiency. (Analyze)
CO4	Develop HPC applications by leveraging HPC frameworks, workload and resource managers, HPC Profilers and debuggers. (Analyze)
CO5	Construct power-aware HPC applications and architecture to reduce the carbon footprint of data centers without major adverse effects on the performance metrics. (Analyze)

**Course Articulation Matrix**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M	M								S	S	S
CO2	S	S	S	M	M								S	S	S
CO3	S	S	S	M	M								S	S	S
CO4	S	S	S	M	M								S	S	S
CO5	S	S	S	M	M	S	S	S					S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

### **Detailed Syllabus:**

**Parallel Processing fundamentals;** Levels and model of parallelism: instruction, transaction, task, thread, memory, function, data flow models, demand-driven computation.

**Parallel architectures:** superscalar architectures, multi-core, multi-threaded, server and cloud;

**Fundamental design issues in HPC:** Load balancing, scheduling, synchronization and resource management.

Operating systems for scalable HPC

**Parallel languages and programming environments;** OpenMP, Pthread, MPI, java, Cilk; Performance analysis of parallel algorithms; Fundamental limitations in HPC: bandwidth, latency and latency hiding techniques.

**Benchmarking HPC:** scientific, engineering, commercial applications and workloads;

**Scalable storage systems:** RAID, SSD cache, SAS, SAN;

HPC based on cluster, cloud, and grid computing: economic model, infrastructure, platform, computation as service;

**Accelerated HPC:** architecture, programming and typical accelerated system with GPU, FPGA, Xeon Phi, Cell BE;

**Power-aware HPC Design:** computing and communication, processing, memory design, interconnect design, power management;

**Assorted topics:** peta scale computing; big data processing, optics in HPC, quantum computers.

### **Reading List:**

1. Georg Hager and Gerhard Wellein. Introduction to High Performance Computing for Scientists and Engineers (1st ed.). CRC Press, Chapman & Hall/CRC Computational Science, India, 2010.
2. Vipin Kumar, Ananth Grama, Anshul Gupta, George Karypis. Introduction to Parallel Computing, 2nd ed., Pearson India, 2003.
3. John L. Hennessy and David A. Patterson. Computer Architecture: A Quantitative Approach (5th ed.). Elsevier India Pvt. Ltd. 2011.
4. David B. Kirk and Wen-mei W. Hwu. Programming Massively Parallel Processors: A Hands-On Approach (1st ed.). Elsevier India Pvt. Ltd. 2010.

CS3881	Game Theory	DEC	3-0-0	3 Credits
--------	-------------	-----	-------	-----------

**Pre-requisites:**

- i. Probability, Statistics and Stochastic Processes (MA2031)
- ii. Modelling and Optimization Techniques (EC2511)
- iii. Differential and Integral Calculus (MA1011)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Analyze games based on complete and incomplete information about the players
<b>CO2</b>	Analyze games where players cooperate
<b>CO3</b>	Compute Nash equilibrium
<b>CO4</b>	Apply game theory to model network traffic
<b>CO5</b>	Analyze auctions using game theory

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	L	M	S	S	S		S			S			S	S	S
<b>CO2</b>	M	L	M	M	L			S		S		S	S	S	S
<b>CO3</b>	M	S	L	M	M	S	S		M		S	S	M	M	S
<b>CO4</b>	M	M	M	S	M	S	S		S	S			M	S	S
<b>CO5</b>	M	L	M	L	L	S		S	S			S	M	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Noncooperative Game Theory:** Games in Normal Form - Preferences and utility, examples of normal-form, Analyzing games: Pareto optimality, Nash equilibrium, Max-min and minmax strategies, dominated strategies, Rationalizability, Correlated equilibrium

**Computing Solution Concepts of Normal-Form Games:** Computing Nash equilibria of two-player, zero-sum games, Computing Nash equilibria of two-player, general-sum games, Complexity of computing Nash equilibrium, Lemke–Howson algorithm, Searching the space of supports, Computing Nash equilibria of n-player, general-sum games, Computing max-min and minmax strategies for two-player, general-sum games, Computing correlated equilibria

**Games with the Extensive Form:** Perfect-information extensive-form games, Subgame-perfect equilibrium, Computing equilibria, Imperfect-information extensive-form games, Sequential equilibrium

**Other Representations:** Repeated games: Finitely repeated games, Infinitely repeated games, automata, Stochastic games Bayesian games: Computing equilibria

**Coalitional Game Theory:** Transferable Utility, Analyzing Coalitional Games, Shapley Value, the Core

**Mechanism Design:** Strategic voting, unrestricted preferences, Implementation, quasilinear setting, efficient mechanisms, and Computational applications of mechanism design, Task scheduling, Bandwidth allocation in computer networks

**Auctions:** Single-good auctions, Canonical auction families, Bayesian mechanisms, Multiunit auctions, combinatorial auctions

#### **Reading List:**

- 1) Shoham, Y. and Leyton–Brown, K. Multiagent Systems: Algorithmic, Game Theoretic, and Logical Foundations. Cambridge University Press, 2008.
- 2) Osborne, M. J., and Rubinstein, A. A Course in Game Theory. Cambridge, MA: MIT Press, 1994.
- 3) D. Fudenberg and J. Tirole, Game Theory, The MIT Press, 2005.

<b>CS3891</b>	<b>PROGRAM ANALYSIS AND VERIFICATION</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--	------------	------------------	------------------

This course introduces students to the theoretical foundations and techniques for automatically analyzing and verifying programs to uncover errors and to assert correctness of programs.

**Pre-requisites:**

- i. Computer Networks (CS3031)
- ii. Compiler Design (CS2071)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Infer properties of a program through automatic analysis techniques. (Analyze)
CO2	Prove correctness of programs using techniques such as abstract interpretation. (Analyze)
CO3	Infer bugs in programs by applying suitable program analysis techniques. (Analyze)
CO4	Prove security related properties of a program using suitable analysis and verification techniques. (Analyze)
CO5	Infer suitable test cases for automatic verification of programs. (Analyze)

**Course Articulation Matrix:**

PO/ PSO CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M	M								S	S	S
CO2	S	S	S	M	M								S	S	S
CO3	S	S	S	M	M								S	S	S
CO4	S	S	S	M	M								S	S	S
CO5	S	S	S	M	M								S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

## Detailed Syllabus:

**Program Representations:** ASTs, control-flow graphs, stack machines.

### Program Semantics

**Data-Flow Analysis:** theoretical framework, sample analyses, soundness and correctness, termination and complexity, computing an over-approximation of join-over-all-paths information using Kildall's algorithm, by modeling the statements in the program as a set of equations. Analysis of multi-procedure programs.

**Inter-procedural Data-Flow Analysis:** context-sensitivity, precision, termination.

### Control-Flow Analysis for Functional Languages

**Pointer Analysis:** Andersen and Steengaard's analysis, field sensitivity for object-oriented languages, call graph construction with dynamic dispatch.

**Hoare logic:** axiomatic semantics, Hoare triples, strongest postconditions and weakest preconditions, proving properties of programs.

**Abstract Interpretation:** Lattices, abstract join-over-all-paths analysis of a program. Correctness of abstract information: Galois connections, abstract interpretation as an over-approximation of concrete semantics.

**Type Systems:** Monomorphic type systems. Pointer analysis of imperative programs. Program slicing. Assertion reasoning using Hoare logic.

### Software Model Checking

**Satisfiability Modulo Theories:** SAT, SMT, DPLL.

### Symbolic Execution and Concolic Testing

**Fuzz Testing:** white-box and grey-box analysis.

**Dynamic Analysis:** examples for concurrency, software engineering applications.

### Program Synthesis

### Program Repair

## Reading List:

1. Nielson, Nielson, and Hankin. Principles of Program Analysis, Springer, 1999.
2. Course Notes on Program Analysis – CMU. Available at: <https://cmu-program-analysis.github.io/2022/resources/program-analysis.pdf>
3. Parametric shape analysis via 3-valued logic. Sagiv, Reps, and Wilhelm, TOPLAS 2002. Available at: <http://www.cs.wisc.edu/wpis/papers/toplas02.pdf>

4. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. Cousot and Cousot. POPL 1977. Available at: <http://www.di.ens.fr/~cousot/publications.www/CosotCousot-POPL-77-ACM-p238--252-1977.pdf>
5. Program analysis via graph reachability. Reps, 1998. Available at: <http://www.cs.wisc.edu/wpis/papers/tr1386.pdf>
6. Saturn: A scalable framework for error detection using Boolean satisfiability. Xie and Aiken, TOPLAS. Available at: [http://theory.stanford.edu/~aiken/publications/papers/saturn\\_toplas.pdf](http://theory.stanford.edu/~aiken/publications/papers/saturn_toplas.pdf)
7. The ASTREE Static Analyzer, Available at: <http://www.astree.ens.fr/>
8. Determining bounds on execution times, Wilhelm. Available at: <https://www.cis.upenn.edu/~alur/CIS670/TimingAnalysisChapter.pdf>
9. CheckFence: Checking consistency of concurrent data types on relaxed memory models. Available at: <https://www.cis.upenn.edu/~alur/PLdi07.pdf>
10. CUTE: A concolic unit testing engine for C; Sen, Marinov, Agha; FSE 2005. Available at: <https://www.cis.upenn.edu/~alur/CIS670/cute.pdf>
11. Logics and automata for software model checking. Alur and Chaudhuri, 2006. Available at: <https://www.cis.upenn.edu/~alur/Marktoberdorf06.pdf>
12. Research papers.

<b>CS3902</b>	<b>Deep Learning Lab</b>	<b>DEC</b>	<b>0-1-2</b>	<b>2 Credits</b>
---------------	--------------------------	------------	--------------	------------------

**Pre-requisites:**

- i. Machine Learning (CS3011)
- ii. Artificial Intelligence (CS2081)

**Course Outcomes:** At the end of the course, students will be able to:

CO1	Make use of deep learning APIs like Keras
CO2	Implement multiple conversions for Analysis
CO3	Apply deep learning techniques for object identification and segmentation
CO4	Implement RNN and CNN for multiple problems
CO5	Implement Autoencoders and GAN.

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	M	L						L			S	S	S
CO2	S	M	L	L						L			S	S	S
CO3	S	M	M	L						L			S	S	S
CO4	S	M	M	L						L			S	S	S
CO5	S	M	M	L						L			S	S	S

**List of Experiments:**

1. Setting up the Spyder IDE Environment and Executing a Python Program.
2. Feed forward neural network for prediction of logic gates.
3. Object detection using image labeling tools.
4. Time-Series Forecasting with the LSTM Model.
5. Conversion of speech to text and vice-versa.
6. Applying the Convolution Neural Network on computer vision problems.
7. Predict a caption for a sample image using CNN, LSTM.
8. Character recognition using CNN and RNN.
9. Image classification on MNIST dataset (CNN model with Fully connected layer).
10. Applying the Deep Learning Models in the field of Natural Language Processing.

11. Train a sentiment analysis model on IMDB dataset, use RNN layers with LSTM/GRU.
12. Applying the Autoencoder algorithms for encoding the real-world data.
13. Develop Autoencoders using MNIST Handwritten Digits.
14. Applying Generative Adversarial Networks for image generation and unsupervised tasks.

### **Reading List:**

- 1) Navin Kumar Manaswi, Deep Learning with Applications Using Python
- 2) Chatbots and Face, Object, and Speech Recognition with TensorFlow and Keras, Apress,2018.
- 3) Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, MIT Press,2016.
- 4) Josh Patterson and Adam Gibson, “Deep learning: A practitioner's approach”, O'Reilly Media, First Edition, 2017.

## IV Year I Semester Department Electives

CS4401	Privacy Preserving Data Processing	DEC	3-0-0	3 Credits
--------	------------------------------------	-----	-------	-----------

### Pre-requisites:

- i. Cryptography and Network Security (CS3061)
- ii. Data Science (CS3411)

**Course Outcomes:** At the end of this course, students will be able to:

CO1	Identify and demonstrate risks to privacy in data science settings,
CO2	Correctly match differential privacy technology with an application,
CO3	Safely implement privacy solutions and experimentally validate the performance and utility of algorithms.
CO4	Understand differential privacy at a level sufficient to engage in research about best practices in implementation, apply the material in practice, and/or connect it to other areas
CO5	Formulate and carry out an interesting, short-term independent research project, and present the work in both written and oral form.

### Course Articulation Matrix:

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M	L									S	S	S
CO2	S	M	L	L									S	S	S
CO3	S	M	L	L									S	S	S
CO4	S	M	L	L									S	S	S
CO5	S	M	L	L									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

### Detailed Syllabus:

**Introduction:** Why Data Privacy? Privacy definitions (k-anonymity,  $\ell$ -diversity).

**Foundation of Differential Privacy:** Formal definition, Properties: composition, invariance, postprocessing,  $(\epsilon, \delta)$ -DP.

**Local Privacy Models:** Differential privacy with untrusted curator, Randomized Response, Heavy hitter detection.

**Convex Optimization:** Basics of optimization, convex functions, Strong convexity, Lipschitz continuous, Gradient Descent

**Relaxations of Differential Privacy:** Moment accountant, Concentrated Differential Privacy, Renyi Differential Privacy, Concentration of measure, Chernoff bound, Hoeffding's inequality

**Statistical inference and machine learning under differential privacy:** Law & policy considerations, Tiered access models, Side-channel & randomness attacks on implementations, Combining differential privacy and cryptography

**Reading List:**

**1)** The Algorithmic Foundations of Differential Privacy, Cynthia Dwork and Aaron Roth. DOI: 10.1561/04000000042.

**2)** The Algorithmic Foundations of Differential Privacy by Cynthia Dwork and Aaron Roth, Publisher: Now Publisher, ISBN: 978-1-60198-818-8

**3)** Differential Privacy: A Primer for a Non-technical Audience by Nissim et al., Harvard University Privacy Tools Project

**4)** Elements of Statistical Learning, Hastie et al., Springer

**5)** Convex Optimization, Boyd and Vandenberghe, Cambridge

<b>CS4411</b>	<b>Human Computer Interaction</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-----------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Computer Graphics (CS3451)

**Course Outcomes:** At the end of the course the student will be able to:

CO1	To develop an awareness of the range of general human-computer interaction issues that must be considered when designing information systems (Apply)
CO2	To analyse interaction problems from a technical, cognitive, and functional perspective (Apply)
CO3	To develop multimodal displays for conveying and presenting information (Apply)
CO4	To design and evaluate the quality of user interfaces and spatial displays (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	L	L	L	L	L	L							S	S	S
CO2	L	L	L	L	L	L							S	S	S
CO3	L	L	L	L	L	L							S	S	S
CO4	L	L	L	L	L	L							S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Introduction:** Usability of Interactive Systems- Usability goals and measures, usability motivations, universal usability, goals for Professional Managing design processes: Introduction, Organizational design to support usability, Four pillars of design, Development methodologies, Ethnographic observation, Participatory design, Scenario Development, Social impact statement for early design review, legal issues, Usability Testing and Laboratories.

**Design:** Introduction, Task- Related Menu Organization, Single menus, Combinations of Multiple Menus, Content Organization, Fast Movement Through Menus, Data entry with Menus: Form Fill-in, dialog Boxes, and alternatives, Audio Menus and menus for Small Displays.

**Command and Natural Languages:** Command organization Functionality, Strategies and Structure, Naming and Abbreviations, Natural Language in Computing Interaction Devices: Introduction, Keyboards and Keypads, Pointing Devices, Speech and Auditory Interfaces, Displays- Small and large.

**Quality of Service:** Models of Response-Time impacts, Expectations and attitudes, User Productivity, Variability in Response Time, Frustrating Experiences Balancing Function and Fashion: Introduction, Error Messages, Nonanthropomorphic Design, Display Design, Web Page Design, and Window Design. Documentation and Information Search: Online Vs Paper Documentation, reading from paper Vs from Displays, Shaping the content of the Documentation, Accessing the Documentation,

**Online tutorials and animated documentation,** online communities for User Assistance, The Development Process, Searching in Textual Documents and Database Querying, Multimedia Document Searches.

### **Reading List:**

- 1) Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven M Jacobs,
- 2) Designing the User Interface, Strategies for Effective Human Computer Interaction, 5ed, Pearson
- 3) Wilbert O Galitz, The Essential guide to user interface design,2/e, Wiley DreamaTech.

<b>CS4421</b>	<b>Advanced Computational Statistics</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--	------------	------------------	------------------

Students of this course upon completion will gain a broad comprehension of the importance of computation in statistics and machine learning. Focus of this course will be on the mathematical and statistical underpinnings of why and how seminal modern statistical methods and inference works.

**Pre-requisites:**

- i. Probability, Statistics and Stochastic Processes (MA2031)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Apply theorems and probability axioms in constructing statistical models and inferring their parameters. (Apply)
CO2	Apply CDF estimation techniques, hypothesis testing, and Bayesian inference in performing inference with respect to parametric and non-parametric models. (Apply)
CO3	Construct statistical models using stochastic processes with a comprehension of their underlying representational ability in modeling the uncertainty in the problem domain. (Apply)
CO4	Apply stochastic optimization techniques and Bayesian modeling / inference techniques with a comprehension of how randomness and uncertainty in the domain is modeled. (Apply)
CO5	Illustrate Monte Carlo methods with an understanding of the role of various random sampling strategies in solving a given problem. (Apply)

**Course Articulation Matrix**

PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L										S	S	S
CO2	S	M	L										S	S	S
CO3	S	M	L										S	S	S
CO4	S	M	L										S	S	S
CO5	M	M	L										M	M	M

S: Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

**Probability:** Review: Probability (Sample spaces, Conditional probability, independent events, Bayes' theorem). Review: Random variables (Distribution and probability functions, Discrete and continuous random variables, Bivariate, Marginal and conditional distributions, Multivariate distributions and IID samples, Expectation, Variance, Covariance, Conditional expectation, Moment generating functions).

**Probability Inequalities,** Convergence types of Random variables, Law of large numbers, Central limit theorem, The Delta method, L1 Convergence.

**Statistical Inference:** Parametric and non-parametric models, Point estimation, Confidence sets, Hypothesis testing – a review.

**CDF estimation,** Statistical functionals, Bootstrap: Simulation, Variance estimation, Confidence intervals, Percentile intervals.

**Parametric inference:** Method of moments, Maximum likelihood estimators: Properties, Consistency and Equivariance, Asymptotic normality, Optimality, Multiparameter models, Parametric bootstrap, Sufficiency, Exponential families and Conditional maximum likelihood estimators. Case study on Noise Contrastive Estimation.

**Hypothesis testing and p-values:** Wald test, Chi-square distribution, Pearson's Chi-square test for multinomial data, Permutation test, Likelihood ratio test, Multiple testing, Goodness-of-fit tests.

**Bayesian Inference:** Functions of parameters, Simulation, Large sample properties of Bayes' procedures, Flat priors, improper priors and non-informative priors, Multiparameter problems, Bayesian testing, Strengths and weaknesses of Bayesian inference.

**Statistical Decision theory:** comparing risk functions, Bayes estimators, Minmax rules, Admissibility, Stein's paradox.

Introduction to Variational Inference.

Statistical Processes and Methods:

**Stochastic Processes:** Markov processes, Poisson processes, Birth-death processes.

**Stochastic optimization:** Robbins-Monro and Kiefer-Wolfowitz algorithms, simulated annealing, stochastic gradient methods.

**Simulation methods:** Monte Carlo methods: Rejection sampling, importance sampling, variance reduction methods (Rao-Blackwellization, stratified sampling). MCMC methods: Gibbs sampling, Metropolis-Hastings, Langevin methods, Hamiltonian Monte Carlo, slice sampling. Implementation issues: burnin, monitoring convergence.

Sequential Monte Carlo (particle filtering)

**Reading List:**

- 1) Larry Wasserman, All of Statistics, First Edition, Springer-Verlag, 2004.
- 2) Geof H. Givens and Jennifer A. Hoeting, Computational Statistics, Second Edition, Wiley, 2005.
- 3) Christian P. Robert, George Casella, Monte Carlo Statistical Methods, Springer, First Edition, 2004.

<b>CS4431</b>	<b>REAL TIME SYSTEMS</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--------------------------	------------	------------------	------------------

This course introduces students to design issues and techniques in developing real time operating systems (RTOS) and exposes students to several case studies involving commercial RTOS.

**Pre-requisites:**

- i. Operating Systems (CS2021)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Construct and evaluate components of real-time OS with a comprehension of the limitations and design trade-offs. (Analyze)
CO2	Construct and analyze real-time task scheduling algorithms for uniprocessors and multi-processors. (Analyze)
CO3	Construct and analyze techniques for achieving real-time communication on-top of different networks. (Analyze)
CO4	Build real-time databases to meet the requirements and degree of concurrency expected by the given real-time application. (Analyze)
CO5	Analyze the given real-time OS, network or database with the help of benchmarks using suitable performance metrics. (Analyze)

**Course Articulation Matrix**

PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M	M								S	S	
CO2	S	S	S	M	M								S	S	
CO3	S	S	S	M	M								S	S	
CO4	S	S	S	M	M								S	S	
CO5	S	S	S	M	M								S	S	

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Introduction to Real-Time systems:** Basic concepts, applications of Real-Time systems, basic model of Real-Time systems, characteristics of Real-Time systems, types of Real-Time systems: hard, firm, soft, timing constraints, modeling timing constraints.

**Real-Time task scheduling:** Basic concepts, clock driven scheduling, table driven scheduling, cyclic, schedulers, hybrid schedulers, event driven scheduling, EDF Scheduling, RMA, DMA, resource sharing among RT tasks, Priority inversion, Priority Inheritance Protocol, Highest Locker Protocol, Priority Ceiling Protocol, Scheduling Real-Time tasks in multiprocessor and distributed systems, Fault tolerant scheduling of tasks, clocks in distributed Real-Time systems.

**Commercial Real-Time Operating Systems:** Time services, Features of real-time operating systems, UNIX and Windows as RT OS, POSIX, PSOS, VRTX, QNX, RT Linux, Lynx, other RT OS, benchmarking RT OS, Real-Time OS: OS services, I/O subsystem, Network OS.

**RT communications:** QoS framework, models, Real-Time Communication in a LAN, IEEE 802.4, RETHER, Communication over Packet Switched Networks, Routing algorithms, RSVP, rate control.

**RT databases:** Applications, characteristics of temporal data, Concurrency control, Commercial RT databases.

Recent research topics or papers in Real-Time systems.

#### **Reading List:**

- 1) Rajib Mall, Real-Time Systems, Pearson, 2007.
- 2) J. W. S. Liu, Real-time Systems, Pearson Education, 2008.
- 3) C. M. Krishna and K. G. Shin, Real-Time Systems, McGraw Hill, 2004.
- 4) P. A. Laplante, Real-Time Systems Design and Analysis, Willey, 2004

<b>CS4441</b>	<b>Soft computing</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-----------------------	------------	------------------	------------------

Pre-requisites:

- i. Probability, Statistics and Stochastic Processes (MA2031)

**Course Outcomes:** At the end of the course, the student will be able to:

<b>CO1</b>	Develop a model for the system using fuzzy modeling and analyze a few paradigms for making decisions within a fuzzy environment. (Analyze)
<b>CO2</b>	Identifying the underlying structure in data using Fuzzy Classification and Fuzzy Pattern Recognition (Apply)
<b>CO3</b>	Apply the mechanics of the binary and continuous Genetic Algorithm to optimization problems. (Apply)
<b>CO4</b>	Apply Natural Optimization like simulated annealing, ant colony optimization, and evolutionary strategies to optimization problems. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	S	S										S	M	S
<b>CO2</b>	S	S	S										S	M	S
<b>CO3</b>	S	S	S										S	M	S
<b>CO4</b>	S	S	S										S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Introduction to Soft Computing:** Concept of computing systems, "Soft" computing versus "Hard" computing, Characteristics of Soft computing, Some applications of soft computing techniques

**Fuzzy logic:** Introduction to Fuzzy logic, Fuzzy sets and membership functions, Operations on Fuzzy sets, Fuzzy relations, rules, propositions, implications and inferences, Defuzzification techniques, Fuzzy logic controller design, Some applications of Fuzzy logic.

**Genetic Algorithms:** Concept of "Genetics" and "Evolution" and its application to probabilistic search techniques; Basic GA framework and different GA architectures; GA

operators: Encoding, Crossover, Selection, Mutation, etc; Solving single-objective optimization problems using GAs.

**Multi-objective Optimization Problem Solving:** Concept of multi-objective optimization problems (MOOPs) and issues of solving them; Multi-Objective Evolutionary Algorithm (MOEA); Non-Pareto approaches to solve MOOPs; Pareto-based approaches to solve MOOPs; Some applications with MOEAs.

**Artificial Neural Networks:** Biological neurons and its working; Simulation of biological neurons to problem solving; Different ANNs architectures; Training techniques for ANNs; Applications of ANNs to solve some real-life problems.

### Reading List:

- 1) Fuzzy Logic: A Practical approach, F. Martin, McNeill, and Ellen Thro, AP Professional, 2000.
- 2) Fuzzy Logic with Engineering Applications (3rd Edn.), Timothy J. Ross, Willey, 2010.
- 3) Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering, Nikola K. Kasabov, MIT Press, 1998.
- 4) Fuzzy Logic for Embedded Systems Applications, Ahmed M. Ibrahim, Elsevier Press, 2004.
- 5) An Introduction to Genetic Algorithms, Melanie Mitchell, MIT Press, 2000.
- 6) Genetic Algorithms In Search, Optimization and Machine Learning, David E. Goldberg, Pearson Education, 2002.
- 7) Practical Genetic Algorithms, Randy L. Haupt and sue Ellen Haupt, John Willey & Sons, 2002.
- 8) Neural Networks, Fuzzy Logis and Genetic Algorithms: Synthesis, and Applications, S. Rajasekaran, and G. A. Vijayalakshmi Pai, Prentice Hall of India, 2007.
- 9) Soft Computing, D. K. Pratihar, Narosa, 2008.
- 10) Neuro-Fuzzy and soft Computing, J.-S. R. Jang, C.-T. Sun, and E. Mizutani, PHI Learning, 2009.
- 11) Neural Networks and Learning Machines, (3rd Edn.), Simon Haykin, PHI Learning, 2011.

<b>CS4451</b>	<b>Applications of AI in Healthcare</b>	<b>DEC</b>	<b>2 – 0 – 0</b>	<b>2 Credits</b>
---------------	---	------------	------------------	------------------

Students upon completion of this course will gain a broad comprehension of the role of Computer Science in Healthcare information systems. The course is interdisciplinary in nature and encompasses Healthcare, Computer Science, Entrepreneurship and AI. This course provides a business analytic and entrepreneurial perspective regarding how CS and related technology are leveraged in ideating, designing, implementing, and managing healthcare systems.

**Pre-requisites:**

- i. Artificial Intelligence (CS2081)
- ii. Machine Learning (CS3011)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Illustrate the role of healthcare entrepreneurship in the healthcare industry with an understanding of the fundamentals of healthcare systems and preventive care. (Apply)
CO2	Infer strategies for intellectual property protection and enforcement in healthcare startups with a comprehension of the regulatory, ethical, and legal norms governing healthcare products and systems. (Apply)
CO3	Construct design blueprints for sustainable and high-quality healthcare systems with a comprehension of the role of responsible use of technology and AI in building healthcare solutions. (Apply)
CO4	Construct engineering solutions for medical devices and implants by leveraging data analytics and AI models. (Apply)
CO5	Construct solutions for descriptive, diagnostic, and predictive analytic problems to enable data-driven decision-making in improving healthcare outcomes especially with respect to reaching rural and underserved domains. (Apply)

**Course Articulation Matrix**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L	M		S		S					S	M	M
CO2	S	M	L	M		S		S				S	S	S	S
CO3	S	M	L	M	M	S		S				S	S	S	S
CO4	S	M	L	M	M	S		S				S	S	S	S
CO5	S	M	L	M	M	S		S				S	S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

### **Module 1: Healthcare Entrepreneurship Fundamentals:**

Overview of healthcare entrepreneurship

Understanding the intersection of technology, innovation, and healthcare

Differentiating between product and service in the healthcare sector

Understanding the importance of consumer protection in healthcare

Why B2B and not B2C in the Indian healthcare market

Addressing regulatory, legal, and compliance challenges in healthcare startups

### **Module 2: Fundamentals of Healthcare Systems and Preventive Care:**

Key components of a healthcare system

Primary, secondary, and tertiary levels of healthcare

Understanding the difference between communicable and non-communicable diseases

Investigating the role of technology to improve healthcare affordability.

Overview of healthcare financing models

Financial factors and healthcare access in India

### **Module 3: Intellectual Property and Regulatory Challenges in Healthcare:**

Overview of intellectual property rights and their importance in healthcare innovation

Types of intellectual property: patents, copyrights, and trademarks

Role of intellectual property in protecting healthcare innovations

Understanding the regulatory approval process for medical devices, pharmaceuticals, and digital health solutions

Strategies for intellectual property protection and enforcement in healthcare startups

Ethical and legal standards in healthcare startups

### **Module 4: Responsible AI in Healthcare:**

What is responsible AI?

The role of healthcare professionals in responsible AI

Challenges of gaining trust and acceptance in adopting AI technologies: Liability and accountability and legal responsibilities of healthcare providers.

The ethical challenges of AI in healthcare

Bias in AI

Privacy and data protection in AI

The importance of transparency and accountability in AI

The future of regulation for AI in healthcare

### **Module 5: Biomedical Informatics:**

What is biomedical informatics?

The use of information technology in healthcare

How can data mining and machine learning be used in healthcare?

What are the ethical and legal issues in biomedical informatics?

Privacy and confidentiality

Ownership of data

Informed consent

### **Module 6: Healthcare Technology and Innovation:**

Medical device innovation and development process, needs assessment, prototyping, testing and design thinking.

Understanding end-user needs and pain points.

Strategies for gathering user feedback and incorporating it into design processes.

Mobile health applications and wearable devices

Challenges and opportunities in implementing healthcare technology and innovation.

Ethical considerations and challenges in the adoption of emerging technologies

IoT applications in healthcare monitoring, remote patient care, and chronic disease management

Virtual reality for medical training, simulation, and therapy

Augmented reality in surgical navigation and patient education

Blockchain applications for secure and interoperable healthcare data management

### **Module 7: Introduction to Digital Health:**

Overview of the digital health ecosystem and its key components

Digital Health and their impact on healthcare transformation

Examining the benefits of telemedicine for patients, healthcare providers, and healthcare systems

Analysing the barriers and challenges of implementing telemedicine in rural and underserved areas

Introduction to electronic health records and their role in digital health

Exploring the challenges of EHR implementation and interoperability

Addressing legal and regulatory considerations in telemedicine, including licensure and liability issues

### **Module 8: The Role of Technologists in Designing Medical Devices and Implants: Empowering Computer Science**

Importance of design and engineering in developing safe and effective medical solutions

Regulatory landscape and standards for medical devices in India

Ethical considerations and patient safety in medical device design

Importance of engaging with healthcare professionals and patients in understanding their requirements, the challenges and the desired outcomes of a medical device or implant.

Conducting usability studies and gathering feedback for iterative design improvements

Cost-effectiveness and affordability considerations for Indian healthcare systems

Integration of software and connectivity features in medical devices

Security and privacy considerations in connected healthcare systems

Leveraging data analytics and AI algorithms in medical device design

### **Module 9: Healthcare Analytics**

Definition and importance of healthcare analytics.

Role of data-driven decision-making in improving healthcare outcomes.

Descriptive analytics: summarizing and visualizing healthcare data.

Diagnostic analytics: identifying patterns and correlations.

Predictive analytics: forecasting and modelling future healthcare events.

Prescriptive analytics: optimizing decision-making using predictive models.

Using analytics to personalize treatment.

Challenges and Opportunities in Healthcare Analytics

Ethical and Legal Implications of Healthcare Analytics

Scaling and implementing analytics solutions in healthcare organizations.

#### **Reading List:**

1) Juréen K. The lean healthcare entrepreneur: Is the lean startup methodology applicable when facing a health care challenge?

2) <https://www.diva-portal.org/smash/get/diva2:724197/FULLTEXT01.pdf>

3) Clayton M. Christensen, Jerome H. Grossman, Jason Hwang. The Innovator's

- 4) Prescription: A Disruptive Solution for Health Care. McGraw Hill Professional, 2008
- 5) Pareras, Luis G.. Innovation and Entrepreneurship in the Healthcare Sector: From Idea to Funding to Launch. Greenbranch Publishing, 2011.
- 6) Christensen, Clayton M.. The Innovator's Dilemma: When New Technologies Cause
- 7) Great Firms to Fail. Harvard Business Review Press, 2013.
- 8) Buescher B, Viguerie P. Thriving under disruption: how to succeed in the years ahead.
- 9) McKinsey on Healthcare. <https://healthcare.mckinsey.com/thriving-under-disruption-how-succeed-years-ahead>. Published 2014.
- 10) Topol, Eric. The Patient Will See You Now: The Future of Medicine Is in Your Hands. Basic Books, 2016.
- 11) Gray's Anatomy: The Anatomical Basis of Clinical Practice. Susan Standring Elsevier Limited, 2016.
- 12) Govindarajan, Vijay., Ramamurti, Ravi. Reverse Innovation in Health Care: How to Make Value-based Delivery Work. Harvard Business Review Press, 2018.
- 13) Adam Bohr, Kaveh Memarzadeh Artificial Intelligence in Healthcare. Elsevier Science, 2020.
- 14) Jyotir Moy Chatterjee, Vishal Jain Machine Learning with Health Care Perspective: Machine Learning and Healthcare. Springer International Publishing, 2020.
- 15) Ahmed A. Elngar, Akansha Singh, Krishna Kant Singh, Mohamed Elhoseny. Machine Learning and the Internet of Medical Things in Healthcare. Elsevier Science, 2021.
- 16) Ethics and governance of artificial intelligence for health: WHO guidance. Geneva: World Health Organization; 2021. Licence: CC BY-NC-SA 3.0 IGO
- 17) Gupta, Rajendra Pratap. Digital Health: Truly Transformational. Wolters Kluwer, 2021.
- 18) Sanjeev Kelkar, India's Public Health Care Delivery, Policies for Universal Health Care. Singapore: Palgrave Macmillan, 2021.
- 19) Machine Learning and the Internet of Medical Things in Healthcare. Elsevier Science, 2021.
- 20) Susskind, R. E., Susskind, D. (2022). The Future of the Professions: How Technology Will Transform the Work of Human Experts. Oxford University Press.
- 21) Michael Friebe Novel Innovation Design for the Future of Health: Entrepreneurial Concepts for Patient Empowerment and Health Democratization. Springer International Publishing, 2022.
- 22) Kearney, Claudine. Leading Innovation and Entrepreneurship in Healthcare:
- 23) A Global Perspective. Edward Elgar Publishing, 2022.

- 24) Divya Mishra, Kirti Seth, Pushpa Singh Transformation in Healthcare with Emerging Technologies. (2022). CRC Press.
- 25) Digital Health Transformation with Blockchain and Artificial Intelligence. Chinmay Chakraborty (2022). CRC Press.
- 26) Integrating AI in IoT Analytics on the Cloud for Healthcare Applications. Jaya Mala (2022). IGI Global.
- 27) Trends of Artificial Intelligence and Big Data for E-Health. Houneida Sakly, Jayne Seekins, Kristen Yeom, Moncef Tagina, Mourad Said, Safwan Halabi Springer International Publishing, 2023.
- 28) <https://cdsco.gov.in/opencms/opencms/en/Medical-Device-Diagnostics/Medical-Device-Diagnostics/>

<b>CS4461</b>	<b>Green Computing</b>	<b>DEC</b>	<b>3-0-0</b>	<b>3 Credits</b>
---------------	------------------------	------------	--------------	------------------

This course introduces students to green computing techniques to design architectures and solutions that are energy-aware and focuses on minimizing the carbon footprint of data centers without compromising on the functional and non-functional QoS metrics.

**Pre-requisites: None**

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Infer suitable energy-efficient hardware components to be utilized while orchestrating a compute, storage or network infrastructure. (Analyze)
CO2	Construct and evaluate the performance of software components with an intent of optimizing energy-usage by adopting energy-efficient algorithmic designs. (Analyze)
CO3	Optimize resource management components of a virtualized Cloud Computing environment to minimize energy consumption and to promote optimal resource usage. (Analyze)
CO4	Develop sustainable HPC applications by minimizing the environmental impact of computing while minimizing SLA violations. (Analyze)
CO5	Develop and benchmark power-aware HPC applications using hypervisors and Cloud simulators. (Analyze)

### Course Articulation Matrix

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	M	M	S	S	S					S	S	S
CO2	S	S	S	M	M	S	S	S					S	S	S
CO3	S	S	S	M	M	S	S	S					S	S	S
CO4	S	S	S	M	S	S	S	S					S	S	S
CO5	S	S	S	M	S	S	S	S					S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

### Detailed Syllabus:

**Introduction to Green Cloud Computing:** Migrating into Cloud, Green IT: An Overview, Green Devices and Hardware, Green Software.

**Green Data Centers:** Data Centers and Associated Energy Challenges, Data Centre IT Infrastructure, Data Centre Facility Infrastructure: Implications for Energy Efficiency, IT Infrastructure Management, Green Data Centre Metrics, Data Centre Management Strategies.

**Green Data Storage:** Introduction, Storage Media Power Characteristics, Energy Management for Hard Disks, System-Level Energy Management.

**Green Networks and Communications:** Introduction, Objectives of Green Network Protocols, Green Network Protocols and Standards, Sustainable Information Systems and Green Metrics.

**Green Cloud Computing and Environmental Sustainability:** Energy Usage Model, Features of Clouds Enabling Green Computing, Towards Energy Efficiency of Cloud Computing, Green Cloud Architecture.

**Energy Adaptive Computing for Ecosystem:** Implementing the Data Center Energy Productivity Metric in a High-Performance Computing Data Center, Sustainable Dynamic Application Hosting Across Geographically Distributed Data Centers, Energy Efficient task scheduling and Resources allocation at Data Center using optimization Techniques, Energy Efficient Virtual Machine Provisioning and Migration while minimizing SLA violations.

**Cloud Computing Tools:** Simulators such as CloudSim, iCanCloud, Open Stack, Green Cloud, Open Nebula, Aneka.

Recent Trends in Green Computing.

### **Reading List:**

1. The Green Computing Book: Tackling Energy Efficiency at Large Scale", Edited by Wu-chun Feng, Chapman & Hall/CRC Computational Science, CRC Press, June 2014.
2. "Design Technologies for Green and Sustainable Computing Systems", Edited by Partha Pratim Pande, Amlan Ganguly, Krishnendu Chakrabarty, Springer, 2013.
3. "Harnessing Green IT: Principles and Practices", Edited by San Murugesan, G.R. Gangadharan, Wiley, 2012.
4. "Cloud Computing: Principles and Paradigms", Edited by Rajkumar Buyya, Jams Broberg, Andrzej Goscinski, Wiley, February 2011.

CS4471	AI Ethics and Societal Challenges	DEC	2 – 0 – 0	2 Credits
--------	-----------------------------------	-----	-----------	-----------

**Pre-requisites:**

- i. Artificial Intelligence (CS2081)
- ii. Machine Learning (CS3011)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Explain catastrophic AI risks and their implications for society (Understand).
CO2	Analyze AI architectures, machine learning models, and deep learning paradigms (Analyze).
CO3	Assess AI safety challenges and propose risk mitigation strategies (Evaluate).
CO4	Apply ethical principles to AI development and policymaking (Apply).
CO5	Use game theory concepts to understand AI governance issues (Apply).

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	S	M	L									S	M	L
CO2	S	M	S	M									S	S	M
CO3	S	S	M	S	M								S	S	M
CO4	S	S	S	S	S	M							S	S	S
CO5	S	S	S	S	S	S							S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Overview of Catastrophic AI Risks:** Malicious Use, AI race, Organizational Risks, Rogue AIS, Discussion of connections between risks.

**Artificial Intelligence Fundamentals:** Artificial Intelligence and its Types, Machine Learning and its Types, Deep Learning-Model Building Blocks, Training and Inference, Applications, History and Timeline of Key Architectures.

**Single-Agent Safety:** Monitoring-ML Systems are Opaque, Motivations for Transparency Research, Approaches to Transparency, Emergent Capabilities, Emergent Goal-Directed

Behaviour, Tail Risk: Emergent Goals, Evaluations and Anomaly Detection, Trojan Attacks and Other Security Threats.

**Safety Engineering:** Risk Decomposition, Safe Design Principles, Component Failure Accident Models and Methods, Systemic Factors, Drift into Failure, and Existential Risks.

**Complex Systems:** Introduction To Complex Systems, Complex Systems For AI Safety- Puzzles, Problems, and Wicked Problems, Challenges With Interventionism.

**Ethics and Society:** Beneficial AI and Machine Ethics- Law, Fairness, The Economic Engine, Wellbeing, Preferences, Happiness, Social Welfare Functions, Moral Uncertainty.

**Collective Action Problems:** Game Theory- Game Theory Fundamentals, The Prisoner's Dilemma, The Iterated Prisoner's Dilemma, Collective Action Problems. Conflict- Bargaining Theory, Commitment Problems, Information Problems, Factors Outside of Bargaining Theory

**Reading List:**

1) Hendrycks, D. (2025). Introduction to AI safety, ethics, and society (p. 562). Taylor & Francis.

<b>CS4481</b>	<b>Blockchains</b>	<b>DEC</b>	<b>3-0-0</b>	<b>3 Credits</b>
---------------	--------------------	------------	--------------	------------------

**Pre-requisites:**

- i. Cryptography and Network Security (CS3061)
- ii. Theory and Design of Programming Languages (CS3071)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Familiarize the functional/operational aspects of cryptocurrency eco-system. (Apply)
CO2	Design Smart Contracts for different application scenarios (Apply)
CO3	Analyze different consensus algorithms used in the design of blockchains (Apply)
CO4	Design Blockchain solutions to address Privacy and Anonymity in transaction networks (Apply)
CO5	Implement smart-contract based solutions for different application scenarios using Ethereum (Apply)

Course Articulation Matrix:

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M										S	M	L
CO2	S	M	L										S	L	L
CO3	S	S	M										S	M	L
CO4	S	S	M										S	M	L
CO5	S	S	M										S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Cryptographic basics for cryptocurrency** - a short overview of Hashing, signature schemes, encryption schemes and elliptic curve cryptography

**Bitcoin** - Wallet - Blocks - Merkle Tree - hardness of mining - transaction verifiability - anonymity - forks - double spending - mathematical analysis of properties of Bitcoin.

**Ethereum** - Ethereum Virtual Machine (EVM) - Wallets for Ethereum - Solidity - Smart Contracts - some attacks on smart contracts, Attacks on Ethereum smart contracts, Merkle Trie data structure for Ethereum, Mining in Ethereum, Nakamoto Consensus on permissionless, nameless, peer-to-peer network, Selfish Mining, Known attacks on Bitcoin

**Consensus in Distributed Systems:** The consensus problem - Asynchronous Byzantine Agreement - AAP protocol and its analysis, PBFT, paxos, 2PC, 3PC protocols, RAFT, Honeybadger

**Abstract Models for BLOCKCHAIN** - GARAY model - RLA Model - Proof of Work ( PoW) as random oracle - formal treatment of consistency, liveness and fairness - Proof of Stake ( PoS) based Chains - Hybrid models ( PoW + PoS) .

**Anonymity and Privacy in Block Chains** - Zero Knowledge proofs and protocols in Blockchain - Succinct non interactive argument for Knowledge (SNARK) - pairing on Elliptic curves - Zcash.

#### **Reading List:**

- 1) Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016. (Free download available)
- 2) Joseph Bonneau et al, SoK: Research perspectives and challenges for Bitcoin and cryptocurrency, IEEE Symposium on security and Privacy, 2015 (article available for free download)
- 3) Roger Wattenhofer: The Science of the Blockchain Cengage publishers, 2016
- 4) J.A.Garay et al, The bitcoin backbone protocol - analysis and applications EUROCRYPT 2015 LNCS VOI 9057, ( VOLII ), pp 281-310. (Also available at [eprint.iacr.org/2016/1048](http://eprint.iacr.org/2016/1048))
- 5) R.Pass et al, Analysis of Blockchain protocol in Asynchronous networks, EUROCRYPT 2017, ( [eprint.iacr.org/2016/454](http://eprint.iacr.org/2016/454))
- 6) R Pass et al, Fruitchain, a fair blockchain, PODC 2017 ([eprint.iacr.org/2016/916](http://eprint.iacr.org/2016/916))

<b>CS4491</b>	<b>Intrusion Detection Systems</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	------------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Computer Networks (CS3031)
- ii. Machine Learning (CS3011)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Explain anomaly-based intrusion detection techniques and their applications.
CO2	Apply formal methods for automatic profiling of program behavior to detect anomalies.
CO3	Develop behavior profiles from network traffic data for intrusion detection.
CO4	Perform correlation analysis of intrusion alerts and investigate attack patterns.
CO5	Implement and test an integrated intrusion detection and response system.

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M	L									S	M	L
CO2	S	S	S	M	L								S	S	M
CO3	S	S	S	S	M								S	S	M
CO4	S	S	S	S	S	M							S	S	S
CO5	S	S	S	S	S	S							S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Anomaly-Based Intrusion Detection Systems:** Introduction to anomaly-based IDS, Payload-based vs. Header-based detection approaches, Setting up an Anomaly-Based System (ABS), Case Studies: PAYL and POSEIDON

**Fast Automatic Profiling of Program Behaviour:** Formal specification for intrusion detection, Methodologies for automatic profiling, Case Study: Remus configuration and profiling techniques

**Learning Behavior Profiles from Noisy Sequences:** Learning by abstraction and pattern recognition, Regular expressions and sequence alignment in intrusion detection, Learning algorithms for user profiling, Evaluating artificial traces for security analysis

**Correlation Analysis of Intrusion Alerts:** Techniques for analyzing intrusion alerts, Similarity-based correlation approaches, Predefined attack scenario correlation, Correlation using attack prerequisites and consequences, Privacy considerations in intrusion alert correlation

**Multi-Step Network Attacks and Prevention:** Multi-step attack modeling and analysis, Strategies for hardening networks against multi-step attacks, Methods for correlating and predicting multi-step attacks, Case Studies of advanced persistent threats (APT)

**Bridging Intrusion Alerts and Security Policies:** Security policy formalism for threat response, designing a threat response system, Generating security policies from intrusion alerts

**Integrated Intrusion Detection and Reaction:** Framework for intrusion detection and response, Architectural design for IDS, Intrusion reactions and attack session analysis, Traffic classification and response automation, Testing and evaluation of IDS effectiveness

#### **Reading List:**

- 1) Roberto Di Pietro and Luigi Mancini, Intrusion Detection Systems, Springer
- 2) Rafeeq Ur Rehman, Intrusion Detection Systems with Snort, Pearson Education, Prentice Hall
- 3) Guide to Intrusion Detection and Prevention Systems, National Institute of Science and Technology
- 4) Tim Crothers, Implementing Intrusion Detection Systems: A hands-on guide for Securing the Network

<b>CS4501</b>	<b>Malware Detection and Mitigation</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---	------------	------------------	------------------

**Pre-requisites:**

- i. Operating Systems (CS2021)
- ii. Cryptography and Network Security (CS3061)
- iii. Distributed Computing (CS3051)

Course Outcomes: At the end of the course, the student will be able to:

<b>CO1</b>	Possess the skills to independently analyze modern malware samples using static and dynamic analysis techniques.
<b>CO2</b>	Have an intimate understanding of executable formats, Windows internals and API, and analysis techniques.
<b>CO3</b>	Extract investigative leads from host and network-based indicators associated with a malicious program
<b>CO4</b>	Apply techniques and concepts to unpack, extract, decrypt, or bypass new anti-analysis techniques in future malware samples.

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	M	M	L	L	M								L		
<b>CO2</b>	S	M	M	M	M									M	
<b>CO3</b>	S	S	M	S	S									S	
<b>CO4</b>	S	S	S	S	S						S	S	S		

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Introduction:** Introduction to malware, OS security concepts, malware threats, the evolution of malware, malware types- viruses, worms, rootkits, Trojans, bots, spyware, adware, logic bombs, malware analysis, static malware analysis, dynamic malware analysis.

**Advanced Static Analysis:** X86 Architecture- Main Memory, Instructions, Opcodes and Endianness, Operands, Registers, Simple Instructions, The Stack, Conditionals, Branching, Rep Instructions, C Main Method and Offsets. Analyzing Windows programs, Portable executable file format, and disassembling malicious executable programs. Anti-static analysis techniques- obfuscation, packing, metamorphism, polymorphism.

**Advanced Dynamic Analysis:** Debugging malware- ollydbg, windbg, setting virtual environments- sandboxes, emulators, Hypervisors, virtual machines, live malware analysis, dead malware analysis, analyzing traces of malware- system-calls, api-calls, registries, network activities. Anti-dynamic analysis techniques-anti-vm, runtime-evasion techniques.

**Malware Functionality:** Downloaders, Backdoors, Credential Stealers, Persistence Mechanisms, Privilege Escalation, Covert malware launching- Launchers, Process Injection, Process Replacement, Hook Injection, Detours, APC injection.

**Malware Detection Techniques:** Signature-based techniques: malware signatures, packed malware signatures, metamorphic and polymorphic malware signatures. Non-signature-based techniques: similarity-based techniques, machine-learning methods, invariant inferences.

### **Reading List:**

- 1) Sikorski, Michael, and Andrew Honig. Practical malware analysis: the hands-on guide to dissecting malicious software. No Starch Press, 2012.
- 2) Filiol, Eric. Computer viruses: from theory to applications. Springer Science & Business Media, 2006.
- 3) Ligh, Michael, Steven Adair, Blake Hartstein, and Matthew Richard. Malware analyst's cookbook and DVD: tools and techniques for fighting malicious code. Wiley Publishing, 2010.

CS4511	Advanced Computer Networks	DEC	3 – 0 – 0	3 Credits
--------	----------------------------	-----	-----------	-----------

**Pre-requisites:**

- i. Computer Networks (CS3031)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Explain high-performance switching and routing techniques and their impact on network design (Understand).
CO2	Analyze advanced IP address lookup techniques and packet classification methods (Analyze).
CO3	Evaluate Quality of Service (QoS) mechanisms and traffic management strategies (Evaluate).
CO4	Compare and contrast software-defined networking (SDN) and network function virtualization (NFV) (Evaluate).
CO5	Design and implement programmable networks using SDN, NFV, and network telemetry tools (Apply).
CO6	Assess the role of data center networking and information-centric networking in modern applications (Analyze).
CO7	Investigate emerging trends and future advancements in computer networks (Analyze).

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M	L									S	M	L
CO2	S	S	S	M	L								S	S	M
CO3	S	S	S	S	M								S	S	M
CO4	S	S	S	S	S	M							S	S	S
CO5	S	S	S	S	S	S							S	S	S
CO6	S	S	S	S	S	S							S	S	S
CO7	S	S	S	S	S	S							S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**High-Performance Switching and Routing:** Introduction to high-performance switching and routing, Performance considerations in network design, and IP address lookup mechanisms.

**Advanced IP Address Lookup Techniques:** Trie-based data structures for IP lookup, Optimized trie-based structures, and Hardware implementations of address lookup.

**Packet Classification:** Need for packet classification, Methods and algorithms for packet classification, Implementation challenges, and solutions.

**Traffic Management:** Quality of Service (QoS) mechanisms, Traffic policing and shaping techniques, Differentiated services.

**Network Softwarization:** Introduction to network softwarization, Network virtualization concepts, Emergence and evolution of SDN.

**Software-defined networking (SDN):** Separation of data and control planes, Control plane abstractions, OpenFlow protocol, and applications.

**Network Function Virtualization (NFV):** Concepts, frameworks, and architectures of NFV, High-performance network packet processing, Comparative analysis of NFV and SDN.

**Programmable Networks:** Data plane programmability, Reconfigurable match-action tables, P4 programming language.

**Data Center Networking:** Introduction to data center networking, Characteristics and challenges, Protocol innovations in data centers.

**Network Telemetry and Emerging Trends:** Network telemetry techniques, Serverless computing concepts, SmartNICs, and in-band network telemetry.

**Information-Centric Networking (ICN):** Content distribution in IP networks, Introduction to information-centric networking, Architectural considerations and challenges.

**Future Directions in Networking:** Green and sustainable data centers, Advancements in network softwarization, Prospects and challenges in SDN 3.0.

#### **Reading List:**

- 1) Computer Networking: A Top-Down Approach" – James F. Kurose, Keith W. Ross
- 2) Computer Networks" – Andrew S. Tanenbaum, David J. Wetherall
- 3) High-Performance Switches and Routers" – H. Jonathan Chao, Bin Liu
- 4) SDN: Software Defined Networks" – Thomas D. Nadeau, Ken Gray

<b>CS4521</b>	<b>Internet of Things</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	---------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Computer Networks (CS3031)
- ii. Microprocessors (EC1541)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Analyze the protocol Stack for the Internet of Things to address the heterogeneity in devices and networks
CO2	Develop smart IoT Applications using smart sensor devices and cloud systems
CO3	Development of smart mobile apps for societal applications
CO4	Design secure protocols for IoT systems

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M	L									S	M	L
CO2	S	S	S	M	L								S	S	M
CO3	S	S	S	S	M								S	S	M
CO4	S	S	S	S	S	M							S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

Introduction to IoT architectures, Communication Standards, M2M and Smart IoT Technology, Sensor Solutions and RFIDs in IoT, Data Link Layer Protocols, Network Layer Protocols, Application-oriented communication protocols, Data Acquiring and Storage, Cloud Service models for IoT, Business Models in IoT, Prototyping Embedded Device Software, Web APIs, Hardware platforms, Smart Applications, Mobile app developments for IoT, Machine learning algorithms for IoT Systems, Vulnerabilities, Security Requirements and Threat Analysis, IoT Security Tomography, and Layered Attacker model, Access control methods, Privacy Issues in IoT Architectures, Programming with Raspberry Pi, Data acquisition with a Cloud platform, Implementing smart IoT applications through web APIs.

**Reading List:**

- 1) Arshdeep Bahga, Vijay Madisetti, Internet of Things: A Hands-on Approach, Universities Press, 2015
- 2) Raj Kamal, Internet of Things: Architecture and Design Principles, McGraw Hill Education private limited, 2017
- 3) Kai Hwang, Min Chen, Big Data Analytics for Cloud, IoT and Cognitive Computing, Wiley, 2018

<b>CS4522</b>	<b>Internet of Things lab</b>	<b>DEC</b>	<b>0– 1 –2</b>	<b>2 Credits</b>
---------------	-------------------------------	------------	----------------	------------------

**Pre-requisites:**

- i. Computer Networks (CS3031)
- ii. Microprocessors (EC1541)
- iii. Distributed Computing (CS3051)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Set up and configure IoT devices for data acquisition and communication (Apply).
CO2	Implement and analyze IoT communication protocols at different layers (Analyze).
CO3	Develop cloud-based IoT storage and real-time monitoring applications (Apply).
CO4	Design and implement secure and privacy-aware IoT systems (Apply).

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	M	L									S	M	L
CO2	S	S	S	M	L								S	S	M
CO3	S	S	S	S	M								S	S	M
CO4	S	S	S	S	S	M							S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Lab Experiments**

**Module 1: Introduction to IoT & Communication Protocols**

Setting up the IoT Development Environment – Configuring Raspberry Pi, Arduino, and ESP32 for IoT applications.

Implementing Machine-to-Machine (M2M) Communication – Using MQTT and CoAP protocols for device communication.

RFID-Based IoT Solutions – Implementing an RFID-based smart tracking system.

**Module 2: IoT Network and Communication Protocols**

Data Link Layer Protocols in IoT – Implementing Zigbee, LoRa, and Bluetooth-based IoT networks.

Network Layer Protocols in IoT – Configuring MQTT, CoAP, and HTTP for IoT applications.

Application-Oriented Communication – Developing a REST API for IoT device interaction.

### **Module 3: IoT Data Acquisition & Cloud Integration**

Data Acquisition and Storage – Capturing sensor data and storing it in a cloud database (Google Firebase/AWS IoT/ThingSpeak).

Cloud Service Models for IoT – Configuring cloud services (IaaS, PaaS, SaaS) for IoT applications.

Developing a Business Model for IoT Applications – Case study and prototype development.

### **Module 4: Embedded Software Development for IoT**

Prototyping Embedded Device Software – Writing and deploying firmware for an IoT-enabled microcontroller.

Working with Web APIs – Integrating IoT applications with third-party web services.

Mobile App Development for IoT – Developing an Android/iOS app for IoT device control.

### **Module 5: IoT Security & Privacy**

Identifying IoT Vulnerabilities – Analyzing security threats in IoT systems.

IoT Security Tomography & Layered Attacker Model – Implementing security testing methods.

Access Control in IoT – Implementing authentication and authorization mechanisms for IoT devices.

Privacy Issues in IoT Architectures – Implementing data encryption and privacy-preserving techniques.

### **Module 6: Smart IoT Applications & Advanced Integration**

Programming with Raspberry Pi for IoT Applications – Implementing a real-time sensor monitoring system.

Data Acquisition with a Cloud Platform – Uploading real-time data from IoT devices to the cloud.

Implementing Smart IoT Applications using Web APIs – Developing a fully functional IoT-based automation system.

Final IoT Project – Creating an end-to-end IoT solution integrating sensors, cloud services, and mobile apps.

### **Reading List:**

1) Arshdeep Bahga, Vijay Madisetti, Internet of Things: A Hands-on Approach, Universities Press, 2015

2) Raj Kamal, Internet of Things: Architecture and Design Principles, McGraw Hill Education Private Limited, 2017

3) Kai Hwang, Min Chen, Big Data Analytics for Cloud, IoT and Cognitive Computing, Wiley, 2018.

<b>CS4531</b>	<b>Computer Forensics &amp; Cyber Security</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--	------------	------------------	------------------

**Pre-requisites:**

- i. Operating Systems (CS2021)
- ii. Cryptography and Network Security (CS3061)
- iii. Distributed Computing (CS3051)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Explain fundamental concepts and legal aspects of computer forensics (Understand).
CO2	Apply forensic investigation techniques for data acquisition and recovery (Apply).
CO3	Analyze file systems, memory, and network artifacts for forensic investigations (Analyze).
CO4	Evaluate forensic tools and methodologies in digital investigations (Evaluate).

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	M	L	L	M								L		
CO2	S	M	M	M	M									M	
CO3	S	S	M	S	S									S	
CO4	S	S	S	S	S						S	S	S		

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Foundations of Computer Forensics: Introduction to Computer Forensics;** Definition, Scope, and Importance; Role in Law Enforcement and Human Resources; Professional Methodology and Services; Types of Computer Forensics Technologies; Business, Military and Law Enforcement Applications; Steps in a Forensic Investigation.

**Evidence Handling, Preservation, and Tools: Evidence and Data Capture:** Data Recovery: Backup and Recovery Concepts; Role of Backups in Forensic Context; Evidence Collection and Seizure: Types of Evidence, Collection Options, Obstacles, Rules of Evidence, Volatile Evidence, Chain of Custody; Duplication and Preservation of Digital Evidence, Imaging Techniques, Processing Steps, Legal Considerations; Verification and

Authentication of Evidence; Overview of Forensic Tools: Software and Hardware Tools, Tool Evaluation, Validation, and Testing.

**Specialized Forensics Domains and Crime Scene Processing:** Forensic Analysis and Validation: Data Collection, Data-Hiding Techniques, Validation, Remote Acquisitions; Network Forensics: Live Acquisitions, Standard Procedures, Tools, HoneyNet; Email Forensics: Structure, Investigation, Server/Client Roles; Mobile Device Forensics: Cell Phone Data Acquisition and Analysis; Windows and DOS System Forensics: File Systems, NTFS, Encryption, Registry, Boot Tasks; Processing Incident Scenes: Evidence Handling at Crime Scenes (Private and Law Enforcement), Search Preparation, Evidence Seizure, Hashing, Storage, Reviewing Cases.

**Cybersecurity Principles and Hardware-Level Security:** Introduction to Cybersecurity: CIA Triad (Confidentiality, Integrity, Availability), Threat Models and Risk Management: Cryptography Basics: AES, ECC; Hardware Security Threats: Hardware Trojans, PUFs, Power Analysis Attacks, Fault Attacks; Micro-architectural Attacks: Timing Attacks, Covert Channels, RAM-based Attacks, Cold Boot Attacks.

**Operating System Security and Malware Analysis:** Operating System-Level Attacks: Stack Smashing, Format String, ROP, Heap Attacks, Dynamic Memory Allocation Exploits: Countermeasures, Non-executable Stacks, Canaries, Capability-Based Systems, Malware Analysis Techniques, Side-Channel Attacks in Operating Systems.

### **Reading Lists:**

- 1) Computer Forensics and Investigations by Nelson, Phillips Enfinger, Steuart, CENGAGE Learning.
- 2) Computer Forensics, Computer Crime Investigation by John R. Vacca, Firewall Media, New Delhi.
- 3) "Introduction to Computer Security" by Matt Bishop, Pearson Publications
- 4) Swarup Bhunia and Mark Tehranipoor, Hardware Security: A Hands-on Learning Approach, Elsevier, 2019.
- 5) Michael E. Whitman, Herbert J. Mattord, (2018). Principles of Information Security, 6th edition, Cengage Learning, N. Delhi.

<b>CS4541</b>	<b>Agile Methodologies</b>	<b>DEC</b>	<b>3-0-0</b>	<b>3 Credits</b>
---------------	----------------------------	------------	--------------	------------------

**Pre-requisites:**

- i. Software Engineering (CS3021)

**Course Outcomes:** At the end of the course the student will be able to:

CO1	Apply agile methodology and agile process to create high quality software. (Apply)
CO2	Use Agile methodology for knowledge management. (Apply)
CO3	Apply Agile development and testing techniques to manage risks. (Apply)
CO4	Analyze the pros and cons of working in Agile Team. (Analyze)
CO5	Apply Feature Driven Development on large size projects. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	M	L	M	M	L	L	L	M	M	M	L	S	M	
CO2	M	M	M	S	M	L	L	L	M	M	M	L	S	M	
CO3	M	M	M	S	M	L	L	L	M	M	M	L	S	M	
CO4	M	M	L	S	S	L	L	L	M	M	M	L	M	M	
CO5	M	M	L	M	L	L	L	L	M	M	M	L	S	M	

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Agile Methodology:** Theories for Agile Management, Agile Software Development – Traditional Model vs. Agile Model, Classification of Agile Methods, Agile Manifesto and Principles, Agile Project Management, Agile Team Interactions, Ethics in Agile Teams, Agility in Design, Testing, Agile Documentations, Agile Drivers, Capabilities and Values.

**Agile Process:** Lean Production - SCRUM, Crystal, Feature Driven Development- Adaptive Software Development - Extreme Programming: Method Overview – Lifecycle – Work Products, Roles and Practices.

Agile Knowledge Sharing – Role of Story-Cards – Story-Card Maturity Model (SMM).

**Agility and Requirements Engineering:** Impact of Agile Processes in RE, Current Agile Practices, Variance, Overview of RE Using Agile, Managing Unstable Requirements, Requirements Elicitation, Agile Requirements Abstraction Model, Requirements Management

in Agile Environment, Agile Requirements Prioritization, Agile Requirements Modelling, Generation – Concurrency in Agile Requirements Generation.

**Agility and Quality Assurance:** Agile Product Development, Agile Metrics, Feature Driven Development (FDD), Financial and Production Metrics in FDD, Agile Approach to Quality Assurance, Test Driven Development, Agile Approach in Global Software Development.

### **Reading List:**

- 1) David J. Anderson and Eli Schragenheim, Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results, Pearson, 2003.
- 2) Hazza and Dubinsky, Agile Software Engineering, Springer, 2009th edition, 2008.
- 3) Craig Larman, —Agile and Iterative Development, Addison-Wesley, First Edition, 2003.

<b>CS4551</b>	<b>Big Data Engineering</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-----------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Data Science (CS3411).
- ii. Distributed Computing (CS3051).

**Course Outcomes:** At the end of the course, the students will be able to:

CO1	Analyze Map-Reduce programming model for data-intensive applications (Analyze)
CO2	Build a Big Data computing platform with distributed file system. (Apply)
CO3	Implement optimized join operations for a distributed environment. (Apply)
CO4	Perform exploratory data analysis on a distributed environment. (Apply)
CO5	Construct and evaluate ML models for Big Data. (Apply)
CO6	Implement stream processing applications. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	S	S				M	M		L	S	S	M
CO2	S	S	S	S	S				M	M		L	S	S	M
CO3	S	S	S	S	S				M	M		L	S	S	M
CO4	S	S	S	S	S				M	M		L	S	S	M
CO5	S	S	S	S	S				M	M		L	S	S	M
CO6	S	S	S	S	S				M	M		L	S	S	M

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**Big Data – An Introduction:** Big Data - Definition, an overview of Big Data; HPC & Big Data, Big Data Characteristics, Issues, and challenges of Big Data, Big data Technological approaches and Potential use cases for Big Data.

**MapReduce and the New Software Stack:** Hadoop Framework, Hadoop echo system, Distributed File Systems (HDFS and YARN), MapReduce, Algorithms Using MapReduce: Matrix-vector multiplication, Matrix multiplication, Relational Algebra operations, Join

operations: optimization for join operations, theta join, skew aware join, Extensions to MapReduce, The Communication-Cost Model, Complexity Theory for MapReduce.

**Exploratory Data Analysis:** Visual data analysis techniques, interaction, techniques-Systems and applications.

**In memory processing:** Apache Spark, RDD programming, Spark SQL, Datasets and Dataframes, SparkML.

**Machine Learning for Big Data:** Selecting, Extracting and Transforming features, Basic statistics, ML pipelines, Supervised and Unsupervised models, Model selection and tuning.

**NoSQL:** CAP Theorem, Birth of NoSQL, Key-value databases, Document databases, Graph Databases, Column Databases, Distributed Data bases: Mango DB, Hive, Cassandra.

**Stream processing:** Introduction to Streams Concepts, Stream Data Model and Architecture, Sampling Data in a Stream, Filtering Streams, Algorithms for streams such as Counting Distinct Elements in a Stream, Estimating Moments, Counting Ones in a Window, Decaying Window.

### **Reading List:**

- 1) Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, "Mining of Massive Datasets", Cambridge Universities Press, 3rd ed, 2020.
- 2) David Loshin, "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph", Elsevier, August 23, 2013.
- 3) Tom White, "Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale", Orielly, 4th Edition, 2015.
- 4) Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills, "Advanced Analytics with Spark: Patterns for Learning from Data at Scale", Orielly, 2015.
- 5) Dan Sullivan, "NoSQL for Mere Mortals", Addison-Wesley, 2015
- 6) <https://spark.apache.org/docs/latest/api/python/pyspark.html>
- 7) <https://github.com/adamjshook/mapreducepatterns>
- 8) <https://spark.apache.org/mllib/>

<b>CS4422</b>	<b>Security lab</b>	<b>DEC</b>	<b>0- 1 - 2</b>	<b>2 Credits</b>
---------------	---------------------	------------	-----------------	------------------

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Implement Cryptographic algorithms
CO2	Analyze the network traffic and simulate the attacks
CO3	Design and implement security protocols
CO4	Analyse the protocol using security tools

Course Articulation Matrix:

PO/ PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	S	M		M				L		L	M	L	M	
CO2		S	S	M	S	L			L		L	M	S	S	S
CO3	S	M	S	S	S	L	L	L	S	M	S	S	S	S	S
CO4		S	M	L	S	M	L	L	M	M	S	S	S	L	S

### Detailed syllabus

#### Number Theory and Cryptographic Experiments:

1. Explore precision integer arithmetic by integrating a high-performance numeric library into a cryptographic computation workflow.
2. Use a classic algorithm from number theory to identify relationships between integers in practical cryptographic key operations.
3. Design a method to retrieve modular inverses crucial for cryptographic key recovery and decryption processes.
4. Embed mathematical operations over modular rings into cryptographic routines for secure computations.

5. Simulate polynomial expressions using binary field arithmetic for applications in symmetric key cryptography.
6. Design a text encoding scheme using historical ciphers to analyze the impact of substitution in cryptographic secrecy.
7. Experiment with symmetric key algorithms by simulating block-level transformations over plaintext data.
8. Analyse cryptographic diffusion and confusion by implementing a modern encryption standard with internal transformations.
9. Optimize cryptographic decryption routines using advanced number-theoretic strategies for modular systems.
10. Create a secure communication pipeline using foundational public-key operations derived from number theory.
11. Establish a mutual trust mechanism over insecure channels using an ephemeral key exchange strategy.
12. Investigate message confidentiality using asymmetric encryption with probabilistic techniques.
13. Visualize secure message exchange using geometrically inspired cryptographic primitives.
14. Authenticate data integrity by crafting and validating digital signatures with asymmetric techniques.

### **Network Security Experiments:**

1. Implement proxy re-encryption to design a secure message transfer system without revealing private keys.
2. Implement layered encryption and leader election in a tree-based network for secure document flow.
3. Construct a program for authentication and key management to implement a secure P2P network communication.
4. Generate a program for public-key cryptography to design user-specific encryption/decryption using stored key tables.
5. Design a program for secret sharing and leader-based communication for secure message exchange between two dynamic groups.
6. Implement PGP components in secure email communication.
7. Apply the Kerberos protocol to simulate authentication between a client and a server using tickets and TGS.

### **Security Tools**

1. Capture real-time digital conversations to understand network behaviour and traffic anomalies in unsecured environments.

2. Discover and evaluate potential vulnerabilities in computing systems using attacker-modelled security testing frameworks.
3. Validate the robustness of security protocols by simulating formal analysis in protocol verification environments.
4. Construct custom detection logic to monitor and react to suspicious behaviour in real-time network environments.

**Reading List:**

1. GNU-MP Manual
2. A. Menezes, P. Van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, 2004
3. Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security: Private Communication in a Public World, Prentice Hall, 2002
4. <https://crypto.stanford.edu/abc/>

## Open Electives Syllabus

<b>CS1061</b>	<b>Object Oriented Programming</b>	<b>OEC</b>	<b>2 – 1 – 0</b>	<b>3 Credits</b>
---------------	------------------------------------	------------	------------------	------------------

**Pre-requisites:**

- i. Problem Solving through Computer Programming (CS1011)
- ii. Problem Solving through Computer Programming Lab (CS1012)

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Construct programs using Object Oriented Design principles like abstraction, polymorphism and inheritance and typing. (Apply)
<b>CO2</b>	Develop applications with handlers for user-defined exceptions, according to the given requirements. (Apply)
<b>CO3</b>	Develop efficient multi-threaded applications with synchronization constructs. (Apply)
<b>CO4</b>	Develop interactive GUI applications with event handling to provide rich user experience. (Apply)
<b>CO5</b>	Develop applications that use file input and output. (Apply)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
<b>CO1</b>	S	M	M		L				L			L
<b>CO2</b>	S	M	M	L	L				L			L
<b>CO3</b>	S	M	M	L	L				L			L
<b>CO4</b>	S	M	M	L	M				L			L
<b>CO5</b>	S	M	M	L	M				L			L
<b>CO6</b>	M	S	S	M	M	-	S	-	-	S	-	-

S: Strong correlation, M: Medium correlation, L: Low correlation

## Detailed Syllabus:

**Overview of Object-Oriented Programming and its need**, Java Programming Elements: Classes and Objects, Data types, Constructors, Input-Output Handling, Control structures, Method overloading and overriding, Abstraction and Inheritance, Interfaces, final and static: classes, blocks and methods, Packages.

**Exception Handling**: Types of Exceptions, Exception classes, try, catch, throw, throws and finally, Exception Handling with Method Overriding, Custom Exceptions.

**Multithreaded Programming**: Introduction to multitasking through processes and threads, The Java Thread Model, creating threads, thread life cycle, thread scheduling, thread priorities, daemon thread, synchronization.

**Garbage Collection**, Runtime class and Memory management in Java.

**String handling**: String, StringBuffer, StringBuilder and tokenizer.

**Generics**: The Collections framework: List, Set and Map interfaces, Enumerator.

**Event handling**: Event, Listeners and adapter classes, anonymous inner classes.

**Abstract Windowing Toolkit (AWT)**: Button, Label, Checkbox, CheckboxGroup, TextField, TextArea, Choice, List, Menu, Panel, Scrollbar and Layout managers.

**File I/O**: Character based Streams, Readers and Writers, RandomAccess, Scanner.

## Reading List:

1. Herbert Schildt, Java: The Complete Reference, 11<sup>th</sup> edition, Mc Graw Hill, 2019.
2. Head First Java, Kathy Sierra & Bert Bates, 3<sup>rd</sup> edition, O'Reilly, 2005.
3. Clean Code, Robert C Martin, Pearson, 2012.
4. Timothy Budd, Object Oriented Programming with Java, Updated Edition, Pearson Education, 2020.
5. Object Oriented Programming with Java, Debasis Samanta, IIT Kharagpur, accessed through: <https://cse.iitkgp.ac.in/~dsamanta/java/index.htm>, Accessed on: August 2021.

<b>CS3301</b>	<b>Database Applications</b>	<b>OEC</b>	<b>2-1-0</b>	<b>3 Credits</b>
---------------	------------------------------	------------	--------------	------------------

**Pre-requisites:**

- i. Problem Solving through Computer Programming (CS1011)
- ii. Problem Solving through Computer Programming lab (CS1012)

**Course Outcomes:** At the end of the course, the student will be able to:

CO1	Construct database schema using Database models at conceptual level identifying entities and relationships among entities using E-R and relational models. (Apply)
CO2	Construct database and implement queries using SQL constructs for a given requirement specification. (Apply)
CO3	Construct database design using Normalization and Functional Dependencies to store information without redundancy. (Analyze)
CO4	Design and develop an application using stored procedures for a given requirement specification. (Apply)
CO5	Design and develop an application with database connectivity for a given requirement specification. (Apply)
CO6	Implement database maintenance and control using authorization access control, transaction and concurrency management and recover constructs. (Apply)

**Course Articulation Matrix:**

CO \ PO	Program Outcomes											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	S	M	L	L	S				M	L		
CO2	S	M	L	L	S				M	L		
CO3	S	M	S	L	S				M	L		
CO4	S	M	M	L	S			S	S	M		
CO5	S	M	M	L	S	S	S	S	S	M	L	L
CO6	S	M	M	L	S			S	S	M		

S: Strong correlation, M: Medium correlation, L: Low correlation

## **Detailed Syllabus:**

**Introduction:** Database-System Applications, Purpose of Database Systems, View of Data, Database Languages, Database Design, Database Engine, Database and Application Architecture, Database Users and Administrators, History of Database Systems

**Introduction to the Relational Model:** Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, The Relational Algebra

**Introduction to SQL:** Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Subqueries, Modification of the Database

**Intermediate SQL:** Join Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schemas, Index Definition in SQL, Authorization

**Advanced SQL:** Accessing SQL from a Programming Language, Functions and Procedures, Triggers, Recursive Queries, Advanced Aggregation Features

**Database Design Using the E-R Model:** Overview of the Design Process, The Entity-Relationship Model, Complex Attributes, Mapping Cardinalities, Primary Key, Removing Redundant Attributes in Entity Sets, Reducing E-R Diagrams to Relational Schemas, Extended E-R Features, Entity-Relationship Design Issues, Alternative Notations for Modeling Data

**Relational Database Design:** Features of Good Relational Designs, Decomposition Using Functional Dependencies, Normal Forms, Functional-Dependency Theory, Algorithms for Decomposition Using Functional Dependencies, Decomposition Using Multivalued

**Dependencies,** More Normal Forms, Atomic Domains and First Normal Form, Database-Design Process, Modeling Temporal Data

**Introduction to Indexing,** Query processing and optimization, Transactions, concurrency control and recovery

## **Reading List:**

1) Abraham Silberschatz; Henry F Korth; S Sudarshan, Database System Concepts, 7th Edition, New York, NY: McGraw-Hill Education 2020

2) Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, 2nd Edition, McGraw-Hill, Inc., 2000

<b>CS3311</b>	<b>Web Programming</b>	<b>OEC</b>	<b>2-1-0</b>	<b>3Credits</b>
---------------	------------------------	------------	--------------	-----------------

**Pre-requisites:**

- i. Problem Solving through Computer Programming (CS1011)
- ii. Problem Solving through Computer Programming lab (CS1012)

**Course Outcomes:** At the end of this course, students will be able to:

CO1	Discover the basic web components and development environment. (Apply)
CO2	Design and develop client-side scripting techniques (Apply)
CO3	Design and develop server-side scripting techniques (Apply)
CO4	Build real world applications using client side and server-side scripting languages (Apply)
CO5	Analyze SOAP and RESTful web Services. (Analyze)

**Course Articulation Matrix**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>CO1</b>	S	M	M	L	M								M	L	L
<b>CO2</b>	S	M	S	L	S				M		L		S	M	L
<b>CO3</b>	S	M	S	L	S				M		L		S	M	L
<b>CO4</b>	S	M	S	L	S				S		M		S	M	L
<b>CO5</b>	S	M	S	M	S				M				S	M	L

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed Syllabus:**

**HTML5** - List - Tables - Images - Forms - Frames - Cascading Style sheets, XML Document type definition - XML Schemas, Document Object model.

**Java Script** - Control statements - Functions - Arrays - Objects - Events –Random number Generation - Dynamic HTML with Java Script – Ajax, JSON – Introduction, Responsive Web Design, Front end framework- Angular JS.

**Different kinds of servers** - web servers – Apache & nginx, File servers – time servers- DB servers. Server-side scripting languages Introduction: PHP basics - PHP server applications (No DB).

**Introduction to NodeJS** – asynchronous nature of nodejs – simple apps Integrating application with DB – DB drivers- Integrate NodeJS with NOSQL, Integrate NodeJS with SQL.

**Web Architecture & Web services:** MVC introduction- thin clients Vs Thick clients. Web services – Introduction- SOAP, REST – writing a RESTful service (nodejs + expres). SOAP Vs REST.

### **Reading List:**

- 1) Paul Deitel, Harvey Deitel, Abbey Deitel, “Internet and World Wide Web How to Program”, Pearson, Fifth Edition, 2011.
- 2) Jeffrey C. Jackson, “Web Technologies – A Computer Science Perspective”, Pearson Education, Fourth Edition, 2012.
- 3) Anthony, Accomazzo, Murray Nathaniel, Lerner Ari, “Fullstack React: The Complete Guide to React JS and Friends”, Fullstack.io, 2017.
- 4) Brown, Ethan, “Web Development with Node and Express: Leveraging the JavaScript Stack”, O'Reilly Media, 2019.
- 5) Dayley B., “Node.js, MongoDB, and AngularJS Web Development”, Addison-Wesley Professional, 2014.

<b>CS4301</b>	<b>AI for Engineers</b>	<b>OEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	-------------------------	------------	------------------	------------------

**Pre-requisites:**

**Course Outcomes:** At the end of the course the student will be able to:

CO1	Construct feed forward neural models for real life applications (Apply)
CO2	Construct models using CNN for image and text problems (Apply)
CO3	Construct models using LSTM and GRU for seq-to-seq problems (Apply)
CO4	Construct finetuned models using pretrained models based on Transformers (Apply)

Course Articulation Matrix:

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	S	S	S									S	S	S
CO2	S	S	S	S									S	S	S
CO3	S	S	S	S									S	S	S
CO4	S	S	S	S									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Neural Networks Basics**, Activation functions, Feedforward networks, Training Neural Networks, Optimizers

**Convolutional Neural Networks (CNNs)**, Evolution of CNN Architectures: AlexNet, ZFNet, VGG, InceptionNets, ResNets

**Recurrent Neural Networks (RNNs)**, LSTM, GRU, Encoder Decoder, Attention

**Transformer models**, Self-Attention, Transformer blocks, Parallelizing computations, positional embeddings

**Introduction to Generative AI**, Overview of generative models and their applications, Importance of Generative AI in various domains.

**Introduction to AI Agents:** Introduction to Explainable and Responsible AI, Robustness, Need for Ethics in AI.

**Reading List:**

- 1) Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
- 2) Daniel Jurafsky and James H. Martin. 2025. Speech and Language Processing. 3rd Edition. <https://web.stanford.edu/~jurafsky/slp3/>

<b>CS4311</b>	<b>Numerical Computing with Python</b>	<b>OEC</b>	<b>3 – 0 – 0</b>	<b>3 Credits</b>
---------------	--	------------	------------------	------------------

**Pre-requisites:**

**Course Outcomes:** At the end of the course the student will be able to:

CO1	Use NumPy arrays to store and perform operations on data sets (Apply)
CO2	Generate plots depicting data and model characteristics (Apply)
CO3	Apply Linear Algebra functions on data (Apply)
CO4	Solve Regression problems of real world (Apply)

**Course Articulation Matrix:**

CO \ PO/ PSO	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	L	L	L	L									S	S	S
CO2	L	L	L	L									S	S	S
CO3	L	L	L	L									S	S	S
CO4	L	L	L	L									S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Detailed syllabus:**

**Creating and manipulating NumPy arrays**, Operations using NumPy arrays

**Matplotlib overview**, creating plots with legends and subplots

**Introduction to Pandas**, reading and writing data in Pandas.

**Linear Algebra and Systems of Linear Equations:** Basics of Linear Algebra, Linear Transformations, Systems of Linear Equations, Solutions to Systems of Linear Equations, Solve Systems of Linear Equations in Python

**Eigenvalues and Eigenvectors:** Eigenvalues and Eigenvectors Problem Statement, The Power Method, The QR Method, Eigenvalues and Eigenvectors in Python

**Least Squares Regression:** Least Squares Regression in Python, Least Square Regression for Nonlinear Functions

**Reading List:**

- 1) M. Lutz, Learning Python 5th Edition, O'Reilly Media (2013)
- 2) M. Stewart: Python for Scientists, Cambridge U. Press (2014)
- 3) J. H. Ferziger, Numerical Methods for Engineering Applications, John Wiley & Sons (in TB section).
- 4) <https://pythonnumericalmethods.studentorg.berkeley.edu/notebooks/Index.html>
- 5) [Practical Numerical Computing Using Python](#) : Scientific and Engineering Applications (2021)

<b>CS2301</b>	<b>Foundations of Quantum Computing</b>	<b>PCC</b>	<b>2 – 0 – 0</b>	<b>2 Credit</b>
---------------	---	------------	------------------	-----------------

**Pre-requisites:**

- i) **Introduction to Computational Problem Solving (CS1011)**
- ii) **Introduction to Computational Problem-Solving Lab (CS1012)**

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Illustrate qubit behavior and construct basic quantum circuits using standard quantum gates. (Apply)
<b>CO2</b>	Differentiate between various physical qubit implementations and assess their advantages. (Analyze/Evaluate)
<b>CO3</b>	Discuss challenges in quantum-secure systems and evaluate post-quantum cryptographic methods. (Analyze/Evaluate)
<b>CO4</b>	Interpret concepts in quantum information using operator-sum representation (Apply)

**Course Articulation Matrix:**

<b>PO/ PSO</b>	<b>Program Outcomes</b>												<b>Program Specific Outcomes</b>		
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>CO1</b>	S	M	M		S				M	M			S	S	S
<b>CO2</b>	S	M	M		S				M	M			S	S	S
<b>CO3</b>	S	M	M		S				M	M			S	S	S
<b>CO4</b>	S	M	M		S				M	M			S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Introduction:** Quantum Technologies – four verticals, Motivation for Quantum Technologies

**Basics of qubits:** Review of classical logic gates, Di Vincenzo criteria for realising qubits, Basics of qubit gates and quantum circuits, Physical implementation of qubits, Solid State Qubits, Semiconducting Qubits – quantum dots, spins, Superconducting Qubits – charge, flux and phase, Topological Qubits – proposals and advantages, Atoms and Ions, Trapped

ions, Rydberg atoms, Neutral atoms, Photonic Qubits, Conventional linear optical setups, Integrated Photonics, NMR qubits, Conventional NMR qubits, NV centres

**Overview of applications and recent achievements:** RSA and Shor's algorithm, Quantum Advantage, Long term goals and strategies being followed, Error correction

**Quantum Information Science:** Digital communication and information, quantum information, Decoherence and noise, Kraus operators

**Quantum Computational Complexity:** Qualitative ideas of a Turing machine, Types of Turing machines, Time and Space complexity – P vs NP, PSPACE, Quantum complexity classes – Q, EQP, BQP, BPP, QMA Post Quantum Cryptography (PQC)

**Reading List:**

1. Quantum Information Science – Manenti R., Motta M., 1st Edition, Oxford University Press (2023)
2. Quantum computation and quantum information – Nielsen M. A., and Chuang I. L., 10th Anniversary edition, Cambridge University Press (2010)
3. Elements of Quantum Computation and Quantum Communication, A. Pathak, Boca Raton, CRC Press (2015)
4. An Introduction to Quantum Computing, Phillip Kaye, Raymond Laflamme, and Michele Mosca, Oxford University Press (2006)
5. Quantum computing explained, David McMahon, Wiley (2008)
6. A Pathak, Elements of Quantum Computation and Quantum Communication, Boca Raton, CRC Press (2015)
7. Information Theory, Robert B. Ash, Dover Publications (2003)
8. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage India Pvt. Ltd. (2014)

CS3302	<b>Basic Quantum Computing Programming Lab</b>	<b>DEC</b>	<b>0 – 2 – 2</b>	<b>3 Credit</b>
--------	--	------------	------------------	-----------------

**Pre-requisites:**

- i) **Introduction to Computational Problem Solving (CS1011)**
- ii) **Introduction to Computational Problem-Solving Lab (CS1012)**

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Apply fundamental quantum operations and logic gates to design and simulate simple quantum circuits using appropriate quantum programming tools. (Apply)
<b>CO2</b>	Analyze the behavior of quantum circuits using statevector and measurement outputs to interpret quantum interference, entanglement, and decoherence. (Analyze)
<b>CO3</b>	Analyze and differentiate quantum algorithms with respect to their classical counterparts in terms of efficiency, outcome, and circuit design. (Analyze)
<b>CO4</b>	Design and implement quantum circuits for selected quantum algorithms and execute them on quantum simulators. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M		S				M	M			S	S	S
<b>CO2</b>	S	M	M		S				M	M			S	S	S
<b>CO3</b>	S	M	M		S				M	M			S	S	S
<b>CO4</b>	S	M	M		S				M	M			S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Basics of programming:** Data structures, classes, Object-oriented programming, Data storage and retrieval, Memory allocation Scientific plotting, Simple algorithms and

benchmarking run time: Sorting, Searching, Arithmetic algorithms like GCD, Prime factorisation

**Numerical Integration and differential equations:** Linear 2nd Order ODEs with constant coefficients, Linear 2nd order ODEs with variable coefficients, Boundary value problems, Poisson equation, Laplace equation, Wave equation, Diffusion Equation

**Numerical techniques in linear algebra:** Matrix inverse, Eigenvalue problem, Diagonalisation of matrices, Singular value decomposition, Numerical techniques in Probability and Statistics, (Pseudo) Random number generation, Computing statistical moments for data samples, Least Squares fitting, Error Analysis, Hypothesis Testing, Monte Carlo sampling,

**Quantum Eigen energies of coupled two level systems:** Eigen energies of two-level system coupled to oscillator (Jaynes-Cummings Model), Superposition and Measurement, Entanglement with Bell States, Quantum Teleportation, Driven two-level system – Rabi Problem, Driven damped oscillator — coherent states, Applications to EM theory (e.g. magnetic field simulation)--Electrostatic charge distributions, Magnetostatic current distributions, Finite Element techniques for electromagnetic simulations

Implementing Basic Quantum Gates: Simulate X, Y, Z, H, S, T gates, Implement CNOT, Toffoli, and SWAP gates, Implement Deutsch–Jozsa Algorithm, Grover's Search Algorithm, Quantum Fourier Transform, Variational Quantum Eigen solver

Quantum Machine Learning: Quantum SVM, Custom Feature Maps, Variational Quantum Classifier, Quantum Multi-Class-SVM

**Quantum Graph Problems:** Graph Partitioning with the VQE, Maximal Clique with VQE, Max-Cut with VQE, set packing with VQE, Vertex Cover with VQE

### Reading List:

1. Quantum computation and quantum information – Nielsen M. A., and Chuang I. L., 10th Anniversary edition, Cambridge University Press (2010)
2. Programming Quantum Computers: Essential Algorithms and Code Samples Eric R. Johnston, Nic Harrigan, Mercedes Gimeno-Segovia, Orielly Press
3. Learn Quantum Computing with Python and Q# - Sarah Kaiser Chris Granade, Manning Publications 2020

4. Quantum Computing for Everyone -- Chris Bernhardt MIT Press ISBN: 9780262539531
5. Pub date: September 8, 2020

CS3311	<b>Introduction to Quantum Computation</b>	<b>DEC</b>	<b>3 – 0 – 0</b>	<b>2 Credit</b>
--------	--	------------	------------------	-----------------

**Course Outcomes:** At the end of this course, students will be able to:

<b>CO1</b>	Apply the principles of quantum mechanics to represent and manipulate information using qubits and quantum gates. (Apply)
<b>CO2</b>	Apply basic quantum algorithms to solve computational problems more efficiently than classical methods. (Apply)
<b>CO3</b>	Analyze quantum entanglement and nonlocality through Bell’s theorems and their implications for quantum correlations. (Analyze)
<b>CO4</b>	Analyze the computational complexity of classical and quantum algorithms within the framework of complexity classes. (Apply)

**Course Articulation Matrix:**

PO/ PSO  CO	Program Outcomes												Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
<b>CO1</b>	S	M	M		S				M	M			S	S	S
<b>CO2</b>	S	M	M		S				M	M			S	S	S
<b>CO3</b>	S	M	M		S				M	M			S	S	S
<b>CO4</b>	S	M	M		S				M	M			S	S	S

S: Strong correlation, M: Medium correlation, L: Low correlation

**Qubits versus classical bits:** Spin-half systems and photon polarizations, Trapped atoms and ions, Artificial atoms using circuits, Semiconducting quantum dots, Single and Two qubit gates – Solovay - Kitaev Theorem

**Quantum correlations:** Entanglement and Bell’s theorems

Review of Turing machines and classical computational complexity: Time and space complexity (P, NP, PSPACE)

**Reversible computation,** Universal quantum logic gates and circuits

**Quantum algorithms:** Deutsch algorithm, Deutsch Josza algorithm, Bernstein - Vazirani algorithm, Simon's algorithm

**Database search:** Grover's algorithm

**Quantum Fourier Transform and prime factorization:** Shor's Algorithm.

**Quantum complexity classes** – Q, EQP, BQP, BPP, QMA

**Additional Topics in Quantum Algorithms:** Variational Quantum Eigensolver (VQE), HHL, QAOA

**Introduction to Error correction:** Fault-tolerance, Simple error correcting codes

**Survey of current status: NISQ era processors,** Quantum advantage claims, Roadmap for future

### **Reading List:**

1. Quantum Information Science – Manenti R., Motta M., 1st Edition, Oxford University Press (2023)
2. Quantum computation and quantum information – Nielsen M. A., and Chuang I. L., 10th Anniversary edition, Cambridge University Press (2010)
3. A Pathak, Elements of Quantum Computation and Quantum Communication, Boca Raton, CRC Press (2015)
4. Quantum error correction and Fault tolerant computing, Frank Gaitan, 1st edition, CRC Press (2008)
5. Quantum computing explained, David McMahon, Wiley (2008)
6. Introduction to Quantum Computing: From a lay person to a programmer in 30 steps, Hui Yung Wong, 1st edition, Springer-Nature Switzerland AG (2022)