



AISSMS

**INSTITUTE OF INFORMATION TECHNOLOGY
(I.O.I.T)**

ADDING VALUE TO ENGINEERING

An Autonomous Institute Affiliated to Savitribai Phule Pune University
Approved by AICTE, New Delhi and Recognised by Govt. of Maharashtra
Accredited by NAAC with "A+" Grade | NBA - 5 UG Programmes



ACADEMIC COURSE STRUCTURE [as per NEP]

COMPUTER ENGINEERING

**B.TECH 4 YEAR UG COURSE
(Applicable for the batches admitted from 2022-2023)**

**AISSMS INSTITUTE OF INFORMATION TECHNOLOGY
Kennedy Road, Near RTO,
Pune – 411 001, Maharashtra State, India
Email: principal@aissmsioit.org,
Website: <https://www.aissmsioit.org>**

Institute Vision & Mission

Vision

To be recognized amongst top 10 private engineering colleges in Maharashtra by the year 2026 by rendering value added education through academic excellence, research, entrepreneurial attitude and global exposure.

Mission

- To enable placement of 150 plus students in the 7 lacs plus category & ensure 100% placement of all final year students
- To connect with 10 plus international universities, professional bodies, and organizations to provide global exposure students
- To create conducive environment for career growth, prosperity, and happiness of 100% staff.
- To be amongst top 5 private colleges in Pune in terms of admission cut off.

Department Vision & Mission

Vision

To be known for imparting quality education in computer engineering to serve the changing needs of global industry.

Mission

- To create an environment that fosters technical and professional growth to make graduates globally competent.
- To develop industry ready professionals, researchers, and entrepreneurs to solve real world problems and societal issues.

Program Educational Objectives (PEOs)

Graduate will:

- **PEO1:** Work productively as successful computer professionals / entrepreneurs / researchers in global industry.
- **PEO2:** Adapt latest technological skills to face challenges of the modern computing industry.
- **PEO3:** Work in finance, healthcare, security and banking sectors as a team member or a leader with ethical values

Program Outcomes (POs)

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. [**Engineering knowledge**]
2. Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. [**Problem analysis**]
3. 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. [**Design/development of solutions**]
4. 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. [**Conduct investigations of complex problems**]
5. 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. [**Modern tool usage**]
6. 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. [**The engineer and society**]
7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. [**Environment and sustainability**]
8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. [**Ethics**]
9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. [**Individual and team work**]
10. 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. [**Communication**]
11. 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. [**Project management and finance**]
12. 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. [**Life-long learning**]

Program Specific Outcomes (PSOs)

Graduates will be able to

- **PSO1:** Problem Solving and Programming Skills: apply knowledge of procedural or object oriented programming to solve computing problems
- **PSO2:** Professional Skills: design and develop software systems using front end and back end technologies by following standard software engineering principles
- **PSO3:** Professional Skills: analyze the algorithms and implement optimized solutions in the domain of machine learning, natural language processing, security, cloud computing and Internet of Things.

A. Definition of Credit:

1 Hr. Lecture (L) per week	1 credit
1 Hr. Tutorial (T) per week	1 credit
1Hr. Practical(P)per week	0.5 Credits
2Hrs. Practical(Lab)/week	1 credit

B. Range of credits –

A range of credits from 160 to 176 for a student to be eligible to get Undergraduate degree in Engineering. A student will be eligible to get Undergraduate degree with Honors or additional Minor Engineering, if he/she completes an additional 20 credits.

C. Credit for Undergraduate Degree in Computer Engineering

Sr. No.	Year	Semester	Credits
1	First Year	I	21
2		II	19
3	Second Year	III	22
4		IV	24
5	Third Year	V	23
6		VI	25
7	Final Year	VII	12
8		VIII	14
Total Credits			160

D. Structure of Undergraduate Engineering program

Sr. no.	Domains	Code	Credits	NEP Suggested
1	Basic Science courses	BSC	19	14-18
2	Engineering Science courses	ESC	16	12-16
3	Programme Core Courses	PCC	56	44-56
4	Programme Elective courses	PEC	15	20
5	Open Elective other than particular Programme	OEC	06	08
6	Vocational and Skill Enhancement Course	VSE	08	08
7	Humanities Social Science and Management	HSM	12	14
8	Experiential Learning Courses	ELC	24	22
9	Liberal Learning Courses	LLC	04	04
Total			160	160-176

E. Domain wise Credit distribution :

Sr. No.	Code	Credits									
		Semesters								Total	NEP Suggested
		I	II	III	IV	V	VI	VII	VIII		
1	BSC	8	8	-	3	-	-	-	-	19	14-18
2	ESC	9	7	-	-	-	-	-	-	16	16-12
3	PCC	-	-	16	12	12	12	04	-	56	44-56
4	PEC	-	-	-	-	04	04	07	-	15	20
5	OEC	-	-	03	-	03	-	-	-	06	08
6	VSE	01	01	-	03	-	03	-	-	08	08
7	HSM	-	-	03	03	03	03	-	-	12	14
8	ELC	3	3	--	01	01	01	01	14	24	22
9	LLC	-	-	-	02	-	02	-	-	04	04
Total Credits		21	19	22	24	23	25	12	14	160	-
Exam Total		650	650	725	725	725	725	600	600	5400	-
Total Working Hours per Week		30	28	26	28	27	31	15	16	201	-

F. Honors Course: Prompt Engineering

Sr. No.	Offered in Semester & Course Code	Courses Name	Semester	Hours per week			Credit	Examination Scheme				Total	
				Lecture	Tutorial	Practical		ISE	ESE	TW	PR	OR	
1	COHDT511	PROMPT Engineering Essentials	V	3		2	4	40#	60*	25	-	25	150
2	COHDT613	PROMPT Template Designing	VI	3		2	4	40#	60*	25	-	25	150
3	COHDT707	PROMPT Engineering Applications	VII	3		4	5	40#	60*	25	25	-	150
4	COHDT803	PROMPT Engineering and AI Integration	VIII	3		4	5	40#	60*	25	25	-	150
Total Credits for Honors Course				12	-	12	18	160	240	100	50	50	600

G. BTech Degree with Research Methodology

Sr. No.	Course Code	Courses Name	Semester	Hours per week			Credit	Examination Scheme					Total
				Lecture	Tutorial	Practical		ISE	ESE	TW	PR	OR	
1	COHDR708	Research Methodology	VII	3	-	-	3	40#	60*	-	-	-	100
2	COHDR709	Mathematical Modelling	VII	3	-	-	3	40#	60*	-	-	-	100
3	COHDR710	Dissertation Phase I	VII	-	-	4	2	--	--	25	-	25	50
4	COHDR804	Research Publication & Ethics	VIII	2	-	-	2	20#	30#	--	--	--	50
5	COHDR805	Paper Publication	VIII	-	-	4	2	-	-	50	-	-	50
6	COHDR806	Dissertation Phase II	VIII	--	--	12	6	-	-	100	--	50	150
Total Credits for Honors in RM Course				08	--	20	18	100	150	175	--	75	500

H. Major Courses

Sr. No.	Course Code	Courses Name	Semester	Hours per week			Credit	Examination Scheme					Total
				Lecture	Tutorial	Practical		ISE	ESE	TW	PR	OR	
1	COPCC302	Discrete Mathematics	III	3	--	--	03	40#	60*	--	--	--	100
2	COPCC303	Data Structures & Algorithm	III	3	--	--	03	40#	60*	--	--	--	100
3	COPCC304	Object Oriented Programming	III	3	--	--	03	40#	60*	--	--	--	100
4	COPCC305	Computer Organization & Operating System	III	3	--	--	03	40#	60**	--	--	--	100
5	COPCC307	Data Structures & Algorithm Laboratory	III	--	--	4	02	--	--	25	50	--	75
6	COPCC308	Object Oriented Programming Laboratory	III	--	--	2	01	--	--	--	50	--	50
7	COPCC309	Computer Organization & Operating System Laboratory	III	--	--	2	01	--	--	25	--	--	25
8	COPCC403	Advanced Data Structures and Algorithm	IV	3	--	--	03	40#	60*	--	--	--	100

Program: Computer Engineering

9	COPCC404	Data Communication & Computer Network	IV	3	--	--	03	40 [#]	60*	--	--	--	100
10	COPCC405	Software Engineering and Modeling	IV	3	--	--	03	40 [#]	60**	--	--	--	100
11	COPCC407	Advanced Data Structures and Algorithm Laboratory	IV	--	--	4	02	--	--	25	50	--	75
12	COPCC408	Computer Network Laboratory	IV	--	--	2	01	--	--	--	50	--	50
13	COELC409	Project Development Laboratory - I	IV	--	--	2	01	--	--	--	--	25	25
14	COPCC502	Database Management Systems	V	3	--	--	03	40 [#]	60*	--	--	--	100
15	COPCC503	Web Technology	V	3	--	--	03	40 [#]	60*	--	--	--	100
16	COPCC504	Theory of Computation	V	3	--	--	03	40 [#]	60**	--	--	--	100
17	COPEC505	Professional Elective I	V	3	--	--	03	40 [#]	60*	--	--	--	100
18	COPCC507	Database Management Systems Laboratory	V	--	--	4	02	--	--	--	50	--	50
19	COPCC508	Web Technology Laboratory											
20	COPEC509	Professional Elective Laboratory- I	IV	--	--	2	01	--	--	25	25	--	50
21	COELC510	Project Development Laboratory – II	V	--	--	2	01	--	--	--	--	50	50
22	COPCC602	Machine Learning	VI	3	--	--	03	40 [#]	60*	--	--	--	100
23	COPCC603	Design and Analysis of Algorithms	VI	3	--	--	03	40 [#]	60*	--	--	--	100
24	COPCC604	Embedded System and Internet of Things	VI	3	--	2	04	40 [#]	60**	--	--	--	100
25	COPEC605	Professional Elective II	VI	3	--	--	03	40 [#]	60*	--	--	--	100
26	COPCC607	Laboratory Practice - II	VI	--	--	4	02	--	--	--	50	--	50
27	COPEC608	Professional Elective Laboratory - II	VI	--	--	2	01	--	--	--	50	--	50
28	COELC609	Project Development Laboratory - III	VI	--	--	2	01	--	--	--	--	50	50

Program: Computer Engineering

29	COPCC701	Deep learning	VII	3	--	--	03	40 [#]	60*	--	--	--	100
30	COPEC702	Elective III	VII	3	--	--	03	40 [#]	60*	--	--	--	100
31	COPEC703	Elective IV	VII	3	--	--	03	40 [#]	60*	--	--	--	100
32	COPCC704	Deep Learning Laboratory	VII	--	--	2	01	--	--	50	50	--	100
33	COPEC705	Elective Laboratory - III	VII	--	--	2	01	--	--	--	50	50	100
34	COELC706	Project Stage – I	VII	--	--	2	01	--	--	50	--	50	100
35	COELC801	Internship	VIII	0	12	0	12	--	--	200	--	100	300
36	COELC802	Project Stage II	VIII	0	0	04	02	--	--	200	--	100	300
Total Credits for Major Courses				55	12	44	89	720	1080	600	475	425	3400

I. Minor Course:

Sr. No.	Courses Code	Courses Name	Semester	Hours per week			Credit	Examination Scheme					Total
				Lecture	Tutorial	Practical		ISE	ESE	TW	PR	OR	
1	COMNR301	Object Oriented Programming	III	3	-	-	3	-	75*	-	-	-	75
2	COMNR302	Object Oriented Programming Laboratory	III	-	-	2	1	-	-	25	-	-	25
3	COMNR401	Software Engineering and Modeling	IV	3	-	-	3	-	75*	-	-	-	75
4	COMNR402	Software Engineering and Modeling Laboratory	IV	-	-	2	1	-	-	25	-	-	25
5	COMNR501	Database Management System	V	3	-	-	3	-	75*	-	-	-	75
6	COMNR502	Database Management System Laboratory	V	-	-	2	1	-	-	25	-	-	25
7	COMNR601	Web Technology	VI	3	-	-	3	-	75*	-	-	-	75
8	COMNR602	Web Technology Laboratory	VI	-	-	2	1	-	-	25	-	-	25
Total Credits for Minor Courses				12		08	16	-	300	100	-	-	400

Program: Computer Engineering

J. Open Elective Courses

Sr. No.	Courses Code	Courses Name	Semester	Hours per week			Credits	Examination Scheme					Total
				Lect.	Tut.	Prac.		ISE	ESE	TW	PR	OR	
1	COOEC306	The Joy of Computing Using Python	III	3	--	--	3	40\$	60\$	--	--	--	100
2	COOEC506	Android Programming	V	3	--	--	3	40\$	60\$	--	--	--	100
Total Credits for Open Elective Courses				06	--	-	06	80	120	--	--	--	200

K. Vocational and Skill Enhancement Courses

Sr. No.	Courses Code	Courses Name	Semester	Hours per week			Credit	Examination Scheme					Total
				Lecture	Tutorial	Practic.		ISE	ESE	TW	PR	OR	
3	COVSE406	Programming in JAVA	IV	1	--	4	3	--	--	50	50	--	100
4	COVSE606	Robotics Process Automation	VI	1	--	4	3	--	--	50	50	--	100
Total Credits for VSE Courses				2	--	8	8	--	--	100	100		200

L. Humanities Social Science and Management Courses

Sr. No.	Courses Code	Courses Name	Semester	Hours per week			Credit	Examination Scheme					Total
				Lect.	Tut.	Prac		ISE	ESE	TW	PR	OR	
1	COHSM301	Democracy Election & Governance	III	2	--	--	2	--	--	25	--	25	50
2	COHSM310	Audit Course- 3 Vedic Mathematics	III	1	--	--	1	--	--	25	--	--	25
3	COHSM401	Design Thinking	IV	2	--	-	2	--	--	25	--	25	50
4	COHSM410	Audit Course- 4 Sustainable Development Goals (SDG)	IV	1	--	--	1	--	--	25	--	--	25
5	COHSM501	Intellectual Property Rights	V	2	--	--	2	--	--	25	--	25	50
6	COHSM510	Audit Course – 5 Foreign Language Level 1 (Japanese or German or etc)	V	1	--	--	1	--	--	25	--	--	25

Program: Computer Engineering

7	COHSM601	Seminar and Technical Paper Writing	VI	1	--	2	2	--	--	50	--	--	50
8	COHSM601	Audit Course- 6 Foreign Language Level 2 (Japanese or German or etc)	VI	1	--	--	1	--	--	25	--	--	25
Total Credits for HSM Courses				11	--	2	12	--	--	250	--	50	300

M. Experiential Learning Courses

Sr. No.	Courses Code	Courses Name	Semester	Hours perweek			Credit	Examination Scheme					Total
				Lect.	Tut.	Pr ac		ISE	ESE	TW	PR	OR	
1	COELC409	Project Development Laboratory - I	IV	--	--	2	1	--	--	--	--	25	25
2	COELC509	Project Development Laboratory - II	V	--	--	2	1	--	--	--	--	25	25
3	COELC609	Project Development Laboratory - III	VI	--	--	2	1	--	--	--	--	50	50
4	COELC706	Project Stage – I	VII	--	--	2	1	--	--	50	--	50	100
5	COELC801	National or International Internship / MOOCs/ Foreign University Certification Courses / Entrepreneur Development /Research Experience	VIII	--	12	0	12	--	--	200	--	100	300
6	COELC802	Project Stage – II	VIII	--	0	04	2	--	--	200	--	100	300
Total Credits for ELC Courses				--	12	12	18	--	--	450	--	350	800

N. Liberal Learning Courses

Sr. No.	Courses Code	Courses Name	Semester	Hours perweek			Credit	Examination Scheme					Total
				Lect.	Tut	Prac.		ISE	ESE	TW	PR	OR	
1	COLLC411	Life Long Learning Skills - 1	IV	--	--	--	1	--	--	--	--	--	--
2	COLLC412	Life Long Learning Skills - 2	IV	--	--	--	1	--	--	--	--	--	--
3	COLLC611	Life Long Learning Skills - 3	VI	--	--	--	1	--	--	--	--	--	--
4	COLLC612	Life Long Learning Skills - 4	VI	--	--	--	1	--	--	--	--	--	--
Total Credits for LLC Courses			--	--	--	04	--	--	--	--	--	--	

All the students are required to acquire 2 credits, one each from A. and B. which will have grades as below:

A. Extracurricular Activities:

Sr. No.	Activity	Level	Achievement	Grade	Achievement	Grade	
1.	Sports	Inter collegiate	Participation	P	Prizewinner	C	
		University	Participation	C	Prizewinner	B	
		Zonal	Participation	B	Prizewinner	B+	
		State	Participation	B+	Prizewinner	A	
		National	Participation	A	Prizewinner	A+	
2.	NSS/NCC	International	Participation	A+	Prizewinner	O	
		Camp	Attended	B			
		Camp+5Activities	Attended	B+			
		Camp+10Activities	Attended	A			
		Camp+15Activities	Attended	A+			
3.	Cultural	Camp+20Activities	Attended	O			
		Inter collegiate	Participation	B	Prizewinner	B+	
		State	Participation	B+	Prizewinner	A	
		National	Participation	A	Prizewinner	A+	
4.	Community Engagement	International	Participation	A+	Prizewinner	O	
		Certified by NGO/Authorities with report and geo- tagged photograph	1 Activity		B		
			2Activities		B+		
			3Activities		A		
			4Activities		A+		
5Activities			O				

B. Co-curricular Activities:

Sr. No	Activity	Level	Achievement	Grade	Achievement	Grade
1.	Conference	National	Participation	B	Prizewinner	A
		International	Participation	B+	Prizewinner	A+
		International(Scopus indexing)	Participation	A+	Prizewinner	O
2.	Journal Publication	Non-refereed but recognized and reputed journal/periodical having ISSN number.		B		
		Refereed Journal - As listed by UGC		A		
		Refereed Journals- As listed by Scopus		A+		
		Refereed Journals - As listed by SCI/SCIE		O		
3.	Hackathon		Participation	A+	Prizewinner	O
4.	Professional Body	National	Membership	P	3 rd Prize	A
			Activities/ participation	B	2 nd Prize	A+
			5participations	B+	1 st Prize	O
5.	Internship	1 week	Completed	C		
		2 week	Completed	B		
		3 week	Completed	B+	Sponsored Project	A+
		4 week	Completed	A	Job through internship	O
6.	Entrepreneurs hip	Awareness camp	Attended	A	Product Developed	A+
					Own Startup	O
7.	Project/Technical events	Inter collegiate	Participation	P	Prizewinner	C
		University	Participation	C	Prizewinner	B
		Zonal	Participation	B	Prizewinner	B+
		State	Participation	B+	Prizewinner	A
		National	Participation	A	Prizewinner	A+
		International	Participation	A+	Prizewinner	O

O. Exit Courses:

Sr. No.	Exit Point	Course Name	Code	Hours per week			Credit	Examination Scheme					Total
				Lec	Tut.	Pract.		ISE	ESE	TW	PR	OR	
Certificate Course in Computer Hardware Maintenance													
1	Exit course after F.Y	Computer Hardware & maintenance Lab	COEXC 101	--	--	4	2	--	--	50	--	--	50
		Object Oriented Programming	COEXC 102	--	--	4	2	--	--	50	--	--	50
		Internship	COEXC 103	--	--	8	4	--	--	100	--	--	100
Total				--	--	16	8	--	--	200	--	--	200
Diploma in Web Development													
2	Exit course after S.Y	Database Management Systems	COEXC 201	--	--	4	2	--	--	50	--	--	50
		Web Technology	COEXC 202	--	--	4	2	--	--	50	--	--	50
		Internship	COEXC 203	--	--	8	4	--	--	100	--	--	100
Total				--	--	16	8	--	--	200	--	--	200
B. Voc in UI/UX Design													
3	Exit course after T.Y	UI/UX Design	COEXC 301	--	--	4	2	--	--	50	--	--	50
		Deep Learning	COEXC 302	--	--	4	2	--	--	50	--	--	50
		Internship	COEXC 303	--	--	8	4	--	--	100	--	--	100
Total Credits for Exit Courses				--	--	16	8	--	--	200	--	--	200

SEMESTER WISE STRUCTURES

Computer Engineering – SY BTech (Semester –III)

Sr. No.	Code	Course Title	Hours per week			Credits	Examination Scheme					
			Lecture	Tutorial	Practical		ISE	ESE	TW	PR	OR	Total
1	COHSM301	Democracy Election & Governance	2	--	--	2	--	--	25	--	25	50
2	COPCC302	Discrete Mathematics	3	--	--	3	40 [#]	60*	--	--	--	100
3	COPCC303	Data Structures & Algorithm	3	--	--	3	40 [#]	60*	--	--	--	100
4	COPCC304	Object Oriented Programming	3	--	--	3	40 [#]	60*	--	--	--	100
5	COPCC305	Computer Organization & Operating System	3	--	--	3	40 [#]	60**	--	--	--	100
6	COOEC306	The Joy of Computing Using Python	3	--	--	3	40 ^{\$}	60 ^{\$\$}	--	--	--	100
7	COPCC307	Data Structures & Algorithm Laboratory	--	--	4	2	--	--	25	50	--	75
8	COPCC308	Object Oriented Programming Laboratory	--	--	2	1	--	--	--	50	--	50
9	COPCC309	Computer Organization & Operating System Laboratory	--	--	2	1	--	--	25	--	--	25
10	COHSM310	Audit Course- Vedic Mathematics	1	--	--	1	--	--	25	--	--	25
		Total	18	--	08	22	200	300	100	100	25	725

* **End Semester Examination (ESE)** based on subjective questions.

** **Practical or Activity based Evaluation.**

In Semester Evaluation based on Presentation/Group Discussion/Laboratory Work/Course Project/Home Assignment/Comprehensive Viva Voce/Blog Writing/Case Study/Survey/Multiple-Choice Question (MCQ) examination.

\$ **For MOOCs:** Assignments marks will be converted on the scale of 40 marks.

\$\$ **For MOOCs:** Score of examination conducted by the respective authority of MOOC or Score of ESE Conducted by Institute will be converted on the scale of 60 marks.

MOOC: The joy of computing using Python (https://onlinecourses.nptel.ac.in/noc22_cs31/preview)

Computer Engineering – SY BTech (Semester –IV)

Sr. No.	Code	Course Title	Hours per week			Credits	Examination scheme					
			Lecture	Tutorial	Practical		ISE	ESE	TW	PR	OR	Total
1	COHSM401	Design Thinking @@	2	--	--	2	--	--	25	--	25	50
2	COBSC402	Applied Mathematics and Computational Statistics	3	--	--	3	40 [#]	60*	--	--	--	100
3	COPCC403	Advanced Data Structures and Algorithm	3	--	--	3	40 [#]	60*	--	--	--	100
4	COPCC404	Data Communication & Computer Network	3	--	--	3	40 [#]	60*	--	--	--	100
5	COPCC405	Software Engineering and Modeling	3	--	--	3	40 [#]	60**	--	--	--	100
6	COVSE406	Programming in JAVA @@	1	--	4	3	--	--	50	50	--	100
7	COPCC407	Advanced Data Structures and Algorithm Laboratory @@	--	--	4	2	--	--	25	50	--	75
8	COPCC408	Computer Network Laboratory	--	--	2	1	--	--	--	50	--	50
9	COELC409	Project Development Laboratory - I	--	--	2	1	--	--	--	--	25	25
10	COHSM411	Audit Course- Sustainable Development Goals (SDG)	1	--	--	1	--	--	25	--	--	25
11	COLLC412	Life Long Learning Skills - 1	--	--	--	1	--	--	25	--	--	25
12	COLLC413	Life Long Learning Skills - 2	--	--	--	1	--	--	25	--	--	25
Total			16	--	12	24	160	240	175	150	50	775
13	--	Minor Course (Th)	3	--	--	3	--	75	--	--	--	75
14	--	Minor Course Lab	--	--	2	1	--	--	25	--	--	25
Total			3	--	2	4	--	75	25	--	--	100
Grand Total			19		14	28	160	315	200	150	50	875

* **End Semester Examination (ESE)** based on subjective questions.

** **Practical or Activity based Evaluation.**

In Semester Evaluation based on Presentation/Group Discussion/Laboratory Work/Course Project/Home Assignment/Comprehensive Viva Voce/Blog Writing/Case Study/Survey/Multiple-Choice Question (MCQ) examination.

Note: @@ Passing is mandatory in both the examination heads to gain the credits.

Computer Engineering – TY BTech (Semester –V)

Sr. No.	Code	Course Title	Hours per week			Credits	Examination scheme					
			Lecture	Tutorial	Practical		ISE	ESE	TW	PR	OR	Total
1	COHSM501	Intellectual Property Rights	2	--	--	2	--	--	25	--	25	50
2	COPCC502	Database Management Systems	3	--	--	3	40 [#]	60*	--	--	--	100
3	COPCC503	Web Technology	3	--	--	3	40 [#]	60*	--	--	--	100
4	COPCC504	Theory of Computation	3	--	--	3	40 [#]	60**	--	--	--	100
5	COPEC505	Professional Elective I	3	--	--	3	40 [#]	60*	--	--	--	100
6	COOEC506	Android Programming	3	--	--	3	40 ^{\$}	60 ^{\$\$}	--	--	--	100
7	COPCC507	Database Management Systems Laboratory	--	--	4	2	--	--	--	50	--	50
8	COPCC508	Web Technology Laboratory	--	--	2	1	--	--	25	--	--	25
9	COPEC509	Professional Elective Laboratory- I	--	--	2	1	--	--	25	25	--	50
10	COELC510	Project Development Laboratory - II	--	--	2	1	--	--	--	--	25	50
11	COHSM511	Audit Course - Foreign Language Level 1 (Japanese or German or etc)	1	--	--	1	--	--	25	--	--	25
Total			19	--	08	23	200	300	75	75	75	725

Elective I :

1. Data Science

2. Object Oriented Modeling and Design

3. Data Mining and Data warehousing

4. Information and Cyber Security

5. Cloud Computing

* **End Semester Examination (ESE)** based on subjective questions.

** **Practical or Activity based Evaluation.**

In Semester Evaluation based on Presentation/Group Discussion/Laboratory Work/Course Project/Home Assignment/Comprehensive Viva Voce/Blog Writing/Case Study/Survey/Multiple-Choice Question (MCQ) examination.

\$ **For MOOCs:** Assignments marks will be converted on the scale of 40 marks.

\$\$ **For MOOCs:** Score of examination conducted by the respective authority of MOOC or Score of ESE Conducted by Institute will be converted on the scale of 60 marks.

MOOC: Introduction to Android App Development

<https://elearn.nptel.ac.in/shop/iit-workshops/completed/introduction-to-android-app-development/>

Computer Engineering – TY BTech (Semester –VI)

Sr. No.	Code	Course Title	Hours per week			Credits	Examination scheme					
			Lecture	Tutorial	Practical		ISE	ESE	TW	PR	OR	Total
1	COHSM601	Seminar and Technical Paper Writing	1	--	2	2	--	--	50	--	--	50
2	COPCC602	Artificial Intelligent and Machine Learning	3	--	--	3	40 [#]	60*	--	--	--	100
3	COPCC603	Design and Analysis of Algorithms	3	--	--	3	40 [#]	60*	--	--	--	100
4	COPCC604	Embedded System and Internet of Things	3	--	2	4	40 [#]	60**	--	--	--	100
5	COPEC605	Professional Elective II	3	--	--	3	40 [#]	60*	--	--	--	100
6	COVSE606	Robotics Process Automation	1	--	4	3	--	--	50	50	--	100
7	COPCC607	Laboratory Practice - I	--	--	4	2	--	--	--	50	--	50
8	COPEC608	Professional Elective Laboratory - II	--	--	2	1	--	--	--	50	--	50
9	COELC609	Project Development Laboratory - III	--	--	2	1	--	--	--	--	50	50
10	COHSM610	Audit Course- Foreign Language Level 2 (Japanese or German or etc)	1	--	--	1	--	--	25	--	--	25
11	COLLC611	Life Long Learning Skills - 3	--	--	--	1	--	--	--	--	--	--
12	COLLC612	Life Long Learning Skills - 4	--	--	--	1	--	--	--	--	--	--
Total			15	--	16	25	160	240	125	150	50	725

Elective II :
1. Data Analytics
2. Software Testing and Quality Assurance
3. Information Retrieval System
4. Blockchain Technology
5. Cloud Security

* **End Semester Examination (ESE)** based on subjective questions.

** **Practical or Activity based Evaluation.**

In Semester Evaluation based on Presentation/Group Discussion/Laboratory Work/Course Project/Home Assignment/Comprehensive Viva Voce/Blog Writing/Case Study/Survey/Multiple-Choice Question (MCQ) examination.

Computer Engineering – Final Year BTech (Semester –VII)

Sr. No.	Code	Course Title	Hours per week			Credits	Examination scheme					
			Lecture	Tutorial	Practical		ISE	ESE	TW	PR	OR	Total
1	COPCC701	Deep learning	3	--	--	3	40 [#]	60*	--	--	--	100
2	COPEC702	Elective III	3	--	--	3	40 [#]	60*	--	--	--	100
3	COPEC703	Elective IV	3	--	--	3	40 [#]	60*	--	--	--	100
4	COPCC704	Deep Learning Laboratory	--	--	2	1	--	--	50	50	--	100
5	COPEC705	Elective Laboratory - III	--	--	2	1	--	--	--	50	50	100
6	COELC706	Project Stage – I	--	--	2	1	--	--	50	--	50	100
Total			09	--	6	12	120	180	100	100	100	600

Elective III :	Elective IV:
2. Natural language Processing	2. Web and Social Media Analytics
3. Cyber Law and Forensics	3. Business Intelligence
4. UI/UX Design	4. Kubernetete and Docker

* **End Semester Examination (ESE)** based on subjective questions.

** **Practical or Activity based Evaluation.**

In Semester Evaluation based on Presentation/Group Discussion/Laboratory Work/Course Project/Home Assignment/Comprehensive Viva Voce/Blog Writing/Case Study/Survey/Multiple-Choice Question (MCQ) examination.

Computer Engineering - Final Year BTech (Semester –VIII)

Sr. No.	Code	Course Title	Hours per week			Credits	Examination scheme					
			Lecture	Tutorial	Practical		ISE	ESE	TW	PR	OR	Total
1	COELC801	National or International Internship / MOOCs/ Foreign University Certification Courses / Entrepreneur Development /Research Experience	--	12	--	12	--	--	200 [@]	--	100	300
2	COELC802	Project – Stage – II	--	--	04	2	--	--	200	--	100	300
Total			--	12	04	14	--	--	400	--	200	600

\$ **For MOOCs:** Assignments marks will be converted on the scale of 40 marks.

\$\$ Score of examination conducted by the respective authority of MOOC or Score of ESE Conducted by Institute will be converted on the scale of 60 marks.

@ Marks obtained in two MOOCs will be converted on the scale of 200 marks.

MOOC: Reinforcement Learning (https://onlinecourses.nptel.ac.in/noc22_cs75/preview)

Under COPRO802 select any two Massive Open Online Course (not less than 8 weeks) listed below:

Sr. No.	MOOC Name	Links
1	Spatial Informatics	https://onlinecourses.nptel.ac.in/noc22_cs108/preview
2	Scalable Data Science	https://onlinecourses.nptel.ac.in/noc22_cs105/preview
3	Machine Learning for Earth System Sciences	https://onlinecourses.nptel.ac.in/noc22_cs119/preview

Program: Computer Engineering

Second Year Computer Engineering (2022 Course) Democracy, Election and Governance			
Course Code:	COHSM301	Credit	2
Contact Hours:	2 Hrs/week (L)	Type of Course:	Lecture/Tutorial
Examination Scheme	Term-work 25 marks	Oral 25 marks	

Pre-requisites:

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External /Internal	Marks
1.	Term-work	Internal	25
	Oral	Internal	25

Course Objectives	
1	To introduce the students meaning of democracy and the role of the governance.
2	To help the understand the various approaches to the study of democracy and governance

Course Outcomes : Students will be able to	
301.1	Know the meaning of democracy and the role of the governance in life.
301.2	Understand the various approaches to the democracy and governance

Topics covered:	
UNIT-I: DEMOCRACY-FOUNDATIONANDDIMENSIONS (5 hrs.) a. Constitution of India b. Evolution of Democracy-Different Models c. Dimensions of Democracy-Social, Economic, and Political	
UNIT-II: DECENTRALIZATION (5 hrs.) a. Indian tradition of decentralization b. History of panchayat Raj institution in the lost independence period c. 73 rd and 74 th amendments d. Challenges of caste, gender, class, democracy and ethnicity	
UNIT-III: GOVERNANCE (5 hrs.) a. Meaning and concepts b. Government and governance c. Inclusion and exclusion	
Text books 1. Banerjee- Dube, I. (2014). A history of modern India. Cambridge University Press. 2. Basu, D. D. (1982). Introduction to the Constitution of India. Prentice Hall of	

Program: Computer Engineering

India.

3. Bhargava, R. (2008). Political theory: An introduction. Pearson Education India.
4. Bhargava,R.,Vanaik, A.(2010) Understanding Contemporary India: Critical Perspective. New Delhi: Orient Blackswan.
5. Chandhoke.N.,Proyadardhi.P,(ed)(2009),‘Contemporary India: Economy, Society, Politics’, Pearson India Education Services Pvt. Ltd, ISBN 978-81- 317-1929-9.
6. Chandra, B. (1999). Essays on contemporary India. Har-Anand Publications.
7. Chaterjee, P. (1997). State and Politics in India.
8. Dasgupta.S.,(ed)(2011),‘Political Sociology’, Dorling Kindersley (India) Pvt. Ltd., Licensees of Pearson Education in south Asia. ISBN: 978-317-6027- 7.
9. Deshpande, S. (2003). Contemporary India: A Sociological View, New Delhi: Viking Publication.
10. Guha, R.(2007).India After Gandhi: The History of the World’s Largest Democracy, Harper Collins Publishers, New York.
11. Guha, R. (2013). Gandhi before India. Penguin UK.
12. Jayal. N.G. (2001). Democracy in India. New Delhi: Oxford University Press.
13. Kohli, A.(1990).Democracy and discontent: India's growing crisis of governability. Cambridge University Press.
14. Kohli, A., Breman, J.,& Hawthorn, G. P.(Eds.)(2001).The success of India's democracy (Vol. 6). Cambridge University Press.
15. Kothari,R.(1989).State against democracy: In search of humane governance. Apex Press.
16. Kothari, R. (1970). Politics in India. New Delhi: Orient Blackswan.
17. Kothari, R. (1995). Caste in Indian politics. Orient Blackswan.
18. Sarkar,S.(2001).Indian democracy: the historical in heritage. the Success of India’s Democracy, 23-46.

मराठी संदर्भ ग्रंथ:

१. राही श्रुती गणेश., आवटे श्रीरंजन, (२०१९), ‘आपलं आयकार्ड’, सुहास पळशीकर द युनिक अकॅडमी पब्लिकेशनप्रा.लि.,.
२. व्होरा राजेंद्र., पळशीकर, सुहास.(२०१४). भारतीय लोकशाही अर्थ आणि व्यवहार. पुणे : डायमंड प्रकाशन.
३. सुमंत, यशवंत.(२०१८). प्रा. यशवंत सुमंत यांची तीन भाषणे. पुणे : युनिक अकॅडमी पब्लिकेशन्स प्रा.लि
४. भोळे. भा.ल. (२०१५). भारतीय गणराज्याचे शासन आणि राजकारण. नागपूर: पिंपळापुरे बुक प्रकाशन
५. कसबे. रावसाहेब. (२०१०)डॉ. आंबेडकर आणि भारतीय राज्यघटना. पुणे: सुगावा प्रकाशन

Note: The assessment for the subject shall be based on presentation and reports submission

**Second Year Computer Engineering (2022 Course)
Discrete Mathematics**

CourseCode:	COPCC302	Credit	3
ContactHours:	3Hrs/week (L)	TypeofCourse:	Lecture
Examination Scheme	In-sem. Evaluation 40 Marks	End sem. Examination 60Marks	

Pre-requisites:

- Mathematics

Course assessment methods/tools:

Sr.No.	Course assessment methods/tools	External/Internal	Marks
1.	In-Sem Evaluation	Internal	40
2.	End Semester Examination	External	60

Course Objectives

1	Use and interpret appropriate set, function and relation terminologies and its operations.
2	Learn logic and proof techniques for mathematical representation.
3	Determine number of logical possibilities of events and solve problems based on algebraic systems.
4	Learn the basic properties of graphs, trees and its application.

Course Outcomes: Students will be able to

302.1	Solve the problems of set theory and mathematical induction.
302.2	Apply concepts of propositional calculus for solving problems.
302.3	Apply the concept of counting principles & algebraic system to solve problems.
302.4	Solve problems on relations and functions.
302.5	Make use of Dijkstra's and minimum spanning tree algorithm(Prim's and Kruskal's) to find the shortest path.

Topics covered:

UNIT I: Set Theory and Proofs (6hrs.)

Introduction and significance of Discrete Mathematics, **Sets:** Importance of set, Naïve Set Theory (Cantor's Set Theory), Axiomatic Set Theory, Set Operations, Cardinality of set, Principle of Inclusion and Exclusion. **Types of Sets** – Bounded and Unbounded Sets, Diagonalization Argument, Countable and Uncountable Sets, Finite and Infinite Sets, Countably Infinite and Uncountably Infinite Sets, Power set, **Proofs:** Proof by Mathematical Induction.

UNIT II: Logic and Propositional Calculus (6hrs.)

Propositions, Logical connectives, conditional and Bi-conditional Propositions, Truth Tables, Tautology, Contradiction, Contingency, Logical Equivalence, Application of Propositional Logic- Translating English Sentences, Conjunctive and Disjunctive Normal Forms, Rules of Inference, Predicates and Quantifiers.

UNIT III: Counting Principles, Groups & Rings (6hrs.)

The Basics of Counting, Rule of Sum and Product, Permutations and Combinations, Binomial Coefficients and Identities, Generalized Permutations and Combinations, Algorithms for generating Permutations and Combinations, Algebraic Systems, Groups, Semi Groups, Monoids, Subgroups, Codes and Group codes, Isomorphism and Automorphism, Homomorphism and Normal Subgroups, Ring, Field.

UNIT IV: Relations and Functions (6hrs.)

Relations and their Properties, n-ary relations and their applications, Representing relations , Closures of relations, Equivalence relations, Partial orderings, Partitions, Hasse diagram, Lattices, Chains and Anti-Chains, Transitive closure and Warshall's algorithm. **Functions-** Surjective, Injective and Bijective functions, Identity function, Partial function, Invertible function, Constant function, Inverse functions and Compositions of functions, The Pigeonhole Principle.

UNIT V: Graph Theory (6hrs.)

Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, the handshaking lemma, Single source shortest path- Dijkstra's Algorithm, Planar Graphs, Graph Coloring.

Case Study: Google Map

UNIT VI- Tree (6hrs.)

Introduction, properties of trees, Binary search tree, tree traversal, decision tree, prefix codes and Huffman coding, cut sets, Spanning Trees and Minimum Spanning Tree, Kruskal's and Prim's algorithms, The Max flow- Min Cut Theorem (Transport network).

Case Study: Neural Network

Syllabus contents required for competitive exams (GATE, UPSC, MPSC etc.)(if complete unit is applicable then write only "unit1/2/.." or write the contents from that unit):

Unit No: 1,2,3,4

Text Books:

1. C. L. Liu, —Elements of Discrete Mathematics, TMH, ISBN 10:0-07-066913-9
1. N. Biggs, —Discrete Mathematics, 3rd Edition, Oxford University Press, ISBN 0 –19 850717 – 8

Reference Books:

1. Kenneth H. Rosen, —Discrete Mathematics and its Applications, Tata McGraw-Hill, ISBN 978-0-07-288008-3, 7th Edition.
2. Bernard Kolman, Robert C. Busby and Sharon Ross, —Discrete Mathematical Structures, Prentice-Hall of India /Pearson, ISBN: 0132078457, 9780132078450.
3. NarsinghDeo, —Graph with application to Engineering and Computer Science, Prentice Hall of India, 1990, 0 – 87692 – 145 – 4.
4. Dr. K. D. Joshi, —Foundations of Discrete Mathematics, New Age International Limited, Publishers, January 1996, ISBN: 8122408265, 9788122408263
5. C.D. Cantrell, —Modern Mathematical Methods for Engineers, Cambridge University Press, ISBN-0521670497
6. Eric Gossett, —Discrete Mathematical Structures with Proofs, Wiley India Ltd, ISBN:978-81-265-2758-8

MOOC/ Video Lectures available at:

https://onlinecourses.nptel.ac.in/noc20_cs82/preview

Program: Computer Engineering

Second Year Computer Engineering (2022Course) Data Structures and Algorithm			
Course Code:	COPCC303	Credit	3
Contact Hours:	3Hrs/week (L)	Type of Course:	Lecture
Examination Scheme	In-Sem.Evaluation 40Marks	End-sem. Examination 60Marks	

Pre-requisites: Programming and Problem Solving-I,II

Course assessment methods/tools:

Sr.No.	Course assessment methods/tools	External/Internal	Marks
1.	In-Sem.Evaluation	Internal	40
2.	End Semester Examination	External	60

Course Objectives

1	To introduce the basic concepts of data structures and algorithms.
2	To emphasize various data searching and sorting methods.
3	To implement various data structures like array, linked lists, stacks and queues.
4	To explain about writing algorithms and step by step approach in solving problems with the help of fundamental data structures.
5	To analyze algorithms using space and time complexity.

Course Outcomes: Students will be able to

303.1	Explain the basic concepts of data structures and calculate the time and space complexity of algorithm using step count method.
303.2	Implement array operations and its applications.
303.3	Analyze different searching and sorting algorithms .
303.4	Identify appropriate type of Linked list and develop programs using it.
303.5	Describe and implement operations and applications of stack.
303.6	Make use of Queue linear data structures to implement real world applications.

Topics covered:

UNIT I: INTRODUCTION TO ALGORITHM AND DATA STRUCTURES (6hrs)

Introduction: Data Structures, Abstract Data Types (ADT), Classification (Linear and Non-linear, Static and Dynamic, Persistent and Ephemeral data structures).

Algorithms: Introduction to algorithm, Characteristics of algorithm, Algorithm design tools: Pseudo-code and flowchart.

Complexity of algorithm: Space complexity, Time complexity, Asymptotic notation- Big-O, Theta and Omega, finding complexity using step count method

Program: Computer Engineering

UNIT II: ARRAY (6hrs.) Concept of Sequential Organization, Overview of Array, Operations on Array, Storage Representation and their Address Calculation: Row major and Column Major, Two-dimensional arrays Sparse Matrix: Sparse matrix representation using array, Sparse matrix addition, Transpose of sparse matrix- Simple and Fast Transpose, Time and Space tradeoff. Case Study: Polynomial Operations
UNIT III: SEARCHING AND SORTING (6hrs.) Searching: Search Techniques-Sequential Search/Linear Search, Binary Search String Matching: Introduction, Naïve string matching algorithm, Robin – Karp algorithm, Knuth – Morris – Pratt algorithm Sorting: Types of Sorting-Internal and External Sorting, General Sort Concepts-Sort Order, Stability, Efficiency, and Number of Passes, Comparison Based Sorting Methods-Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Merge sort
UNIT IV: LINKED LIST (6hrs.) Introduction to Static and Dynamic Memory Allocation, Linked List: Introduction, of Linked Lists, Types of Linked List: singly linked, linear and Circular Linked Lists, Doubly Linked List, Doubly Circular Linked List, Primitive Operations on Linked List- Create, Traverse, Search, Insert, Delete, Sort, and Concatenate. Case Study: Addition of long positive integers
UNIT V: STACK (6hrs.) Basic concept, Representation of Stacks Using array and linked list , stack operations Applications of Stack- Recursion, Expression Evaluation and Conversion Case Study: Tower of Hanoi, Backtracking
UNIT VI: QUEUE (6hrs.) Basic concept, Representation of Queue using array and linked list, Queue Operations, Circular Queue and its advantages, Deque -Basic concept, types (Input restricted and Output restricted), Priority Queue. Case Study: Job Scheduling, Website request processing
Syllabus contents required for competitive exams (GATE, UPSC, MPSC etc) All Units are required for GATE exam
Text Books: 1. “Fundamentals of Data Structures in C++”, E. Horowitz, S. Sahni, S. Anderson-freed, University Press, ISBN-10: 8173716064 2. “Data Structures and Algorithms in Python”, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Wiley Publication, ISBN: 978-1-118-29027-9
Reference Books: 1. “C++: The Complete Reference”, Herbert Schildt, The Mc Graw Hill company, ISBN 0- 07- 222680-3 2. “An introduction to data structures with Applications”, Jean-Paul Tremblay, Paul. G. Soresan, Tata Mc-Graw Hill International Editions, ISBN-0-07- 462471-7 3. “Let us Python”, Yashwant Kanetkar & A. Kanetkar, BPB Publisher, ISBN: 9789389845006
MOOC/ Video Lectures available at: Course by Coursera on Ordered data structures https://www.coursera.org/learn/cs-fundamentals-2?action=enroll&authMode=signup&ranEAID=%2AGqSdLGGurk&ranMID=40328&ranSiteID=.GqSdLGGurk-CxpBI2UKW9MKhmU8KjSjdw&siteID=.GqSdLGGurk-CxpBI2UKW9MKhmU8KjSjdw&utm_campaign=%2AGqSdLGGurk&utm_content=10&utm_medium=partners&utm_source=linkshare

Program: Computer Engineering

Second Year Computer Engineering (2022 Course) Object Oriented Programming			
Course Code	COPCC304	Credits	3
Contact Hours.	3 Hrs./Week (L)	Type of Course	Lecture/Practical
Examination Scheme	In Semester (TH): 40 Marks	End Semester (TH): 60 Marks	

Pre-requisites: Programming and Problem Solving-I, II

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External/ Internal	Marks
1.	In Semester Examination	Internal	40
2.	End Semester Examination	External	60

Course Objectives

1	To explore programming skills of students, using object oriented programming concepts
2	To learn the syntax and semantics of the C++ programming language
3	To use the object-oriented paradigm in program development.

Course Outcomes : Students will be able to

304.1	Write a program using different OOP features and types of functions.
304.2	Apply the concept of inheritance to write a program.
304.3	Compare polymorphism techniques and develop a solution for particular problem
304.4	Apply generic programming and exceptional handling concepts to write a program.
304.5	Select appropriate file operations and modes to write a program.
304.6	Illustrate STL Components and write a program using STL components.

Topics covered:

UNIT I : FUNDAMENTAL CONCEPTS OF OOPS

(06 Hours)

Introduction: Introduction to Object Oriented Programming, Object Oriented Paradigm, Features of OOP, benefits of OOP.

Data Types: variables and constants, Class – Data members, Member Functions, and class as abstract data type, Object Visibility Modes, Constructor & Types of Constructors, Destructor, Binding – static & dynamic, Inline Function, Static Members, Static Function, Friend Function, Friend Class, Array of Objects.

Case Study: Demonstrate Class Animal/Car using object, constructor, destructor and functions

UNIT II: INHERITANCE IN C++

(06 Hours)

Derived class & base class, Types of inheritance: Public, Protected and Private Inheritance, Ambiguity in multiple inheritance & multipath inheritance, Constructor & Destructor in Inheritance, Order of Constructor and Destructor Call.

Case Study: Demonstrate Inheritance and its types using Vehicle Hire Company

Program: Computer Engineering

UNIT III: C++ POLYMORPHISM

(06 Hours)

Introduction: Polymorphism – Compile time and Run time Polymorphism, Type conversion, Pointer, Pointers to object, this pointer, Virtual function, Pure virtual function, Abstract class.

Case Study: Demonstrate Online Payment system using function overloading. Demonstrate final amount calculation to be given to fruit Vendor on fruit purchase.

UNIT IV: GENERIC PROGRAMMING & EXCEPTION HANDLING USING C++

(06 Hours)

Generic Programming – Introduction to Template, Types- Function Template, Class Templates Template with multiple parameters.

Exception Handling - Exception Handling: Definition and Types of exceptions, Exception handling using try-catch-throw. Catching mechanism, Exception handling in inheritance.

Separating interfaces and implementation using C++ constructs - class, functions, polymorphism and error handling

Case Study : Demonstrate exception handling in Login Page to any mail sever Like Gmail Demonstrate Ticket Reservation in Railway Reservation system for Adult/Child Passenger

UNIT V: C++ FILE HANDLING

(06 Hours)

File Handling: Classes for file stream operation, Opening and closing a file - File mode, Error Handling functions in file, File Pointers and Their Manipulation, File Operations on Characters, File Operations on Binary Files – Variables, Class Objects, Sequential File Organization.

Case Study: Demonstrate C++ program to redirect output to file using C++ File handling functions

UNIT VI: STANDARD TEMPLATE LIBRARY (08 Hours)

Standard Template Library: Standard Template Library (STL), components of STL: Containers, algorithms and iterators, Containers- Sequence container, associative containers, container adapters, Iterators- input, output, forward, bidirectional and random access, Algorithms- basic searching and sorting algorithms, min-max algorithm, set operations.

Tokenizer using file handling and STL stack.

Case Study: Demonstrate an application for Chair arrangement in a seminar hall using STL Container Vector

OR

Demonstrate an application for students Mark list management using STL map

Syllabus contents required for competitive exams (GATE)

Unit 1,2

Text books

1. E.Balagurusamy, “Object-Oriented Programming with C++”, 7th edition, Graw-Hill Publication, ISBN 10: 9352607996 ISBN 13: 9789352607990
2. Deitel, “C++ How to Program”, 4th Edition, Pearson Education, ISBN:81-297-0276-2

Reference Books:

1. Herbert Scheldt, “C++-The complete reference”, Eighth Edition, McGraw Hill Professional, ISBN:978-00-72226805

MOOC/ Video Lectures available at:

NPTEL Lecture Link

1. Programming in C++
By Prof. Partha Pratim Das | IIT Kharagpur
https://onlinecourses.nptel.ac.in/noc19_cs38/preview

Program: Computer Engineering

Second Year Computer Engineering (2022 Course) Computer Organization and Operating System			
Course Code	COPCC305	Credits	3
Contact Hours.	3 Hrs./Week (L)	Type of Course	Lecture/Practical
Examination Scheme	In Semester(TH): 40 Marks	End Semester(TH): 60 Marks	

Pre-requisites:

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External/ Internal	Marks
1.	In Semester Examination	Internal	40
2.	End Semester Examination	External	60

Course Objectives	
1	To explain the structure, function and characteristics of computer systems.
2	To illustrate the structure and function of I/O and memory Organization.
3	To acquaint the basic of 8086 microprocessor and assembly language.
4	To make aware of different services of operating system and Linux commands

Course Outcomes : Students will be able to	
305.1	Describe functions and components of computer system and compare the architecture of processors.
305.2	Demonstrate the memory partitioning through operating system installations.
305.3	Install input output peripherals.
305.4	Implement 8086 assembly language program.
305.5	Implement process scheduling algorithm .
305.6	Make use of different Linux commands to perform various operations.

Topics covered:

UNIT I:INTRODUCTION TO COMPUTER ORGANIZATION

(6 HOURS)

Computer Organization and Architecture: Concepts of Computer Organization & computer architecture Function and structure components of computer system, Interconnection of components.

Evolution (a brief history) of computers, Designing for Performance, Evolution of Intel processor architecture- 4 bit to 64 bit, performance assessment.

Organization of a processor - Registers, ALU and Control unit, Data path in a CPU, Instruction cycle. Comparison of 8085,8086 and 80386 processors.

Program: Computer Engineering

UNIT II: MEMORY ORGANIZATION (6 HOURS)

Memory Subsystem: Memory cells - SRAM and DRAM cells, Internal Organization of a memory chip, Organization of a memory unit.

Case study: 1.Demonstrate single boot OS installation
2.Demonstrate Dual boot OS installation

UNIT III: I/O ORGANIZATION (6 HOURS)

Input/ Output Subsystem: Access of I/O devices, I/O ports, I/O interfaces - Serial port, Parallel port, PCI bus, SCSI bus, USB bus, I/O peripherals - Input devices, Output devices, Secondary storage devices.

UNIT IV: MICROPROCESSOR (6 HOURS)

Introduction to Microprocessor: Introduction to 16 bit microprocessor, Architecture and Pin diagram of 8086, Programmers model of **8086** (Registers), Addressing modes, Instruction set of 8086 in detail, Instruction Formats, Stacks,

Assembly Language: Assembly Language Programming, Assembler, Linker, Debugger (Turbo debugger), Directives, Loop constructs, 8086 Programming examples.

UNIT V: INTRODUCTION TO OPERATING SYSTEM (6 HOURS)

Operating System Objectives and Functions, The Evolution of Operating Systems, Service of operating system. Operating system services for process management. Process scheduling algorithms.

UNIT VI: INTRODUCTION TO LINUX AND LINUX COMMANDS (6 HOURS)

Introduction to Linux operating system, Linux OS services. Linux distributions and licensing model. **Linux Commands:** Linux Directory Commands, Linux File Commands, Linux File Content Commands, Linux User Commands

Syllabus contents required for competitive exams (GATE)

Unit 1,2,3,5

List of Experiments

1. Assemble the computer system using various hardware components. Study the general structure of computer system.
2. Study of memory hierarchy with respect to capacity, access time, performance and cost per bit.
3. Single boot OS installation (Linux OS installation)
4. Dual boot OS installation (Linux / Windows OS installation)
5. Multi OS installation
6. Demonstrate installation of I/O Peripherals- Printer/Scanner
7. Study of the architecture of 8086 microprocessor
8. Implement 8/16/32/64 bit ALP to accept two digit numbers and display same on command prompt
9. Implement 8/16/32/64 bit ALP to accept two digit numbers perform addition and display result on command prompt.
10. Write a program to implement Scheduling Algorithms: FCFS, SJF and Round Robin.

Program: Computer Engineering

11. Study of various Linux commands.

Text Books:

1. W.Stallings, —Computer Organization and Architecture: Designing for performance, Pearson Education/ Prentice Hall of India, 2003, ISBN 978-93-325-1870-4, 10th Edition.
2. Zaky S, Hamacher, —Computer Organization, 5th Edition, McGraw-Hill Publications, 2001, ISBN- 978-1-25-900537-5, 5th Edition
3. The Design of the Unix Operating System, Maurice J. Bach, Pearson Education, ISBN: 81-7758-770-6
4. Silberschartz, Galvin, Gagne, “Operating System Principles”, 9th Edition, Wiley, ISBN: 978-1-118-06333-0.

Reference Books:

1. The Unix Programming Environment, Brian Kernighan and Rob Pike, Prentice Hall
2. P. Abel, "Assembly Language Programming", 5th edition, Pearson Education, 2002, ISBN 81 - 203 -1037 - 3.

MOOC/ Video Lectures available at:

<https://archive.nptel.ac.in/courses/106/105/106105163/>

<https://archive.nptel.ac.in/courses/106/105/106105163/>

<https://archive.nptel.ac.in/courses/106/105/106105163/>

<https://archive.nptel.ac.in/courses/106/105/106105163/>

<https://spoken-tutorial.org/>

Udemy Course: Linux, Unix essentials and introduction to shell scripting

Program: Computer Engineering

Second Year Computer Engineering (2022 Course)			
The Joy of Computing Using Python			
Course Code:	COOEC306	Credit	3
Contact Hours:	3 Hrs/week (L)	Type of Course:	Lecture
Examination Scheme	In-sem. Evaluation 40 Marks	End-sem.Examination 60 Marks	

Pre-requisites: Problem Solving and Programming

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External/Internal	Marks
1.	In-Sem. Evaluation	Internal	40
2.	End Semester Examination	External	60

Course Objectives

1	The student will have ability to get joyful learning experience through programming.
2	The student will have ability to think logically and arrive at a solution programmatically.
3	To inculcate basic programming skills.

Course Outcomes: Students will be able to

306.1	Write a program using syntax and semantic rules of python.
306.2	Develop gaming applications using list, dictionary.
306.3	Develop gaming applications using tuple.
306.4	Develop logic for competitive programming problems.

Topics covered:

UNIT I: Introduction to Basic Python Programming	(6 hrs.)
Introduction to programming, Introduction to scratch, Introduction to anaconda, Installation of anaconda, Introduction to spyder, Printing statements in python, Understanding variables in python, Executing a sequence of instructions in the console, Writing your first program, Taking inputs from the user, Discount calculation, Motivation to if condition, A reminder on how to deal with numbers, Understanding if condition's working, Realizing the importance of syntax and indentation, Introduction to loops, Introduction to while loop.	
UNIT II: List and Dictionary	(6 hrs.)
List: Introduction, Manipulation, Operations, Slicing, Loops and Conditionals. Dictionary: Introduction, Operations-access items, change items, add items, remove items, Loop dictionaries, Copy dictionaries, Nested dictionaries, dictionary methods. Counting Candies : Crowd to the rescue Permutations : Jumbled Words Birthday Paradox : Find your twin	

Program: Computer Engineering

Spot the similarities : Dabble game

Magic Square Hit & Trial

- Guess the Movie Name

UNIT III: Data Handling Using Dictionary

(6 hrs.)

Dictionaries:

Google Translate : Speak in any Language

Monte Hall : 3 doors and a twist

Rock, Paper and Scissor : Cheating not allowed !!

Sorting : Arrange the books

Searching : Find in seconds

Substitution Cipher : What's the secret !!

Tic tac toe : Let's play

Recursion : Tower of Hanoi

UNIT IV: Data Handling Using Tuple

(6 hrs.)

Tuples- Python Data Structure, Spiral Traversing, GPS-Track the Route, Lottery Simulation, Anagrams

Snakes and Ladders : Down the memory lane

Image Processing : Fun with images

Sentiment Analysis : Analyse your Facebook data

UNIT V- Python Programming: Part-I

(6 hrs.)

Natural Language Processing- Author Stylometry, Introduction to Network, FLAMES, Data Compression

Calculation of the Area : Don't measure

Six degrees of separation : Meet your favorites

UNIT VI- Python Programming: Part-II

(6 hrs.)

Fun with Calendar, Collatz Conjecture

Abstraction Everywhere : Apps in your phone

Page Rank : How Google Works !!

Syllabus contents required for competitive exams (GATE, UPSC, MPSC etc.) NIL

Text Books:

1. Python Programming :Using problem solving approach by Reema Thareja ;oxford
2. Python Programming: A modular approach First Edition by sheetal Taneja Naveen Kumar; Pearson

Reference Books:

1. Learning Python Mark Lutz, David Ascher oreilly
2. Python Object-Oriented Programming: Build robust and maintainable object-oriented Python applications and libraries, 4th Edition Steven F.Lott, Dusty Phillips Packt
3. Beginning Game Development with Python and Pygame: From Novice to Professional (Beginning From Novice to Professional) Will McGugan Apress

MOOC/ Video Lectures available at:

https://onlinecourses.nptel.ac.in/noc22_cs31/preview

Program: Computer Engineering

Second Year Computer Engineering (2022 Course) Data Structures and Algorithm Laboratory			
Course Code:	COPCC307	Credit	2
Contact Hours:	4Hrs/week (P)	Type of Course:	Practical
Examination Scheme	Term Work 25 Marks	Practical 50 Marks	

Pre-requisites: Programming and Problem Solving-I, II

Course assessment methods/tools:

Sr.No.	Course assessment methods/tools	External/Internal	Marks
2.	Practical Examination	External	50
	Term Work	Internal	25

Course Objectives	
1	To understand various data representation techniques.
2	Identify suitable data structure to solve various computing problems.
3	To explain about writing algorithms and step by step approach in solving problems with the help of fundamental data structures.
4	To analyze algorithms using space and time complexity.

Course Outcomes: Students will be able to	
307.1	Select efficient algorithm for solving computational problems.
307.2	Implement various data structures like array, linked lists, stacks and queues.
307.3	Analyze different searching and sorting algorithms using asymptotic notation.
307.4	Identify appropriate type of Linked list and develop programs using dynamic memory management.
307.5	Develop data structure concepts in programming languages like Python,C/C++.

List of Experiments
<p>Set of suggested assignment list is provided in groups- A, B, C, D, and E. Each student must perform at least 11 assignments as at least 2 from group A, 2 from group B, 2 from group C, 2 from group D and 3 from group E.</p> <p>Group A assignments should be implemented in python without using built-in methods for major functionality of assignment. Use List data structure of Python as array. Group B,C, D and E assignments should be implemented in C++ language.</p> <p>Operating System recommended:- 64-bit Open source Linux or its derivative</p> <p>Programming tools recommended: - Open Source python, Programming tool like Jupyter Notebook, Pycharm, Spyder, G++/GCC</p>

Program: Computer Engineering

	Group. A	
	<p>1. You have been given an array of positive integers A_1, A_2, \dots, A_n with length N and you have to print an array of same length (N) where the values in the new array are the sum of every number in the array, except the number at that index. Write program in Python Language.</p>	
	<p>2. Write a python program to compute following computation on matrix:</p> <ol style="list-style-type: none"> a) Addition of two matrices b) Subtraction of two matrices c) Multiplication of two matrices d) Transpose of a matrix <p style="text-align: center;">OR</p> <p>Implement Sparse matrix addition and transpose in C++</p>	
	Group. B	
	<p>3. Implement Linear Search And Binary Search In C/C++/Python. Plot graph of the time taken by both the programs using tool like gnuplot.</p>	
	<p>4. Design and implement an algorithm that will find the top and the least scores of students from an online Quiz. Note: The scores are stored in an array. Expected time complexity is in $O(n \log n)$.</p>	
	<p>5. Implement Insertion sort, Selection sort.</p>	
	Group. C	
	<p>6. Write a C++ program to implement a singly link list and perform operations such as insert, delete, display, search element from it and reverse the list (Find middle node of linked list)</p>	
	<p>7. Write C++ program for storing binary number using doubly linked lists. Write functions- To compute 1's and 2's complement</p>	
	Group. D	
	<p>8. Write a C++ program to evaluate an infix expression, without converting it to postfix. The input string can have spaces, (,) and precedence of operators should be handled.</p>	
	<p>9. Given an expression with only } and { . The expression may not be balanced. Find minimum number of bracket reversals to make the expression balance.</p>	
	Group. E	
	<p>10. Implement Queue using linked list.</p> <p style="text-align: center;">OR</p> <p>You are given an array $A[]$ of size N and an integer K. Your task is to print the minimum element for each subarray of size 'k'. In other words, for each Valid Index i ($0 \leq i \leq N - K$) You have to print $\min(A[i], A[i + 1], A[i + 2] \dots A[i + k])$.</p>	

Program: Computer Engineering

11. Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order once placed cannot be cancelled. Write C++ program to simulate the system using circular queue using array.

Practice Questions

1. Write a Python program to get the length in bytes of one array item in the internal representation
2. Write a Python program to get the number of occurrences of a specified element in an array.
3. Write a Python program to insert a new item before the second element in an existing array.
4. Write a Python program to find whether a given array of integers contains any duplicate element. Return true if any value appears at least twice in the said array and return false if every element is distinct
5. Write a Python program to sort a list of elements using the bubble sort algorithm.
6. Write a Python program to sort a list of elements using the merge sort algorithm.
7. Write a C++ program to find the middle element of a given Linked List.
8. Write a C++ program to find the k^{th} node of a linked list by starting at the middle and moving towards the head.
9. Write a C++ program to implement a singly circular link list and perform operations such as insert, delete, and display.
10. Implement stack using linked list.
11. Implement stack using queues
12. Evaluate postfix expression.
13. Convert decimal to binary number using stack.

Program: Computer Engineering

Second Year Computer Engineering (2022 Course)			
Object Oriented Programming Laboratory			
Course Code	COPCC308	Credits	1
Contact Hours.	2 Hrs./Week (PR)	Type of Course	Practical
Examination Scheme	Practical (50 Marks)		

Pre-requisites: Programming and Problem Solving-I, II

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External/Internal	Marks
1.	Practical Examination	External	50

Course Objectives	
1	To explore programming skills of students, using object oriented programming concepts
2	To learn the syntax and semantics of the C++ programming language
3	To use the object-oriented paradigm in program development.

Course Outcomes : Students will be able to	
308.1	Implement a basic C++ programs.
308.2	Apply the concept of inheritance, polymorphism and file handling to develop a program.
308.3	Develop a program using exceptional handling and generic programming concept.

List of Experiments
<p>Set of suggested assignment list is provided in groups- A, B, and C. Each student must perform at least 8 assignments as at least 3 from group A, 3 from group B, and 2 from group C.</p> <p>Operating System recommended:- 64-bit Open source Linux or its derivative.</p> <p>Programming tools recommended: - G++/GCC, Visual Studio/Eclips/Dev CPP.</p>
Group A
<ol style="list-style-type: none">1. Write a C++ program implement simple Arithmetic Calculator.2. Write a C++ Program<ol style="list-style-type: none">a. Even/ODD number Checkb. Check the character is Vowel or notc. Check Leap Yeard. Create Pyramid and Patterne. Find the largest of 3 entered nos.3. Write a C++ program to perform following operations on an Array<ol style="list-style-type: none">1. Display ODD Indexed Nos.2. Calculate Sum array elements

Program: Computer Engineering

3. Calculate Average of Array Elements
4. Add Or Delete the no. from an array
4. Given an array A of positive integers. Your task is to find the leaders in the array. An element of array is leader if it is greater than or equal to all the elements to its right side. The rightmost element is always a leader
5. Write a C++ program to display Month Name if month no is entered by User using
 1. If- else
 2. Switch case

Compare above 2 conditional control structure

Group B

1. Write a CPP to create class Student with appropriate member variable and member functions and make use of following
 - a. Constructors
 - b. Destructors
 - c. Inline, static, friend function
 - d. Dynamic memory allocation-deallocation
2. Write a CPP to implement following inheritances using car rental system
 - a. Single Inheritance
 - b. Multilevel inheritance
 - c. Multiple Inheritance
 - d. Hierarchical Inheritance
3. Write a CPP to implement Online Payment system using function overloading for Online Shopee.
4. Implement a class Complex which represents the Complex Number data type. Implement the following operations:
 - a. Constructor (including a default constructor which creates the complex number 0+0i).
 - b. Overloaded operator +, - to add and subtract two complex numbers
 - c. Overloaded operator *, / to multiply and divide two complex numbers.
 - d. Overloaded << and >> to print and read Complex Numbers.
5. Write C++ program to calculate monthly and yearly expenses done on Education, Travel, Medical, Misc.

Group C

1. Write C++ Program with base class convert declares two variables, val1 and val2, which hold the initial and converted values, respectively. It also defines the functions getinit() and getconv(), which return the initial value and the converted value. These elements of convert are fixed and applicable to all derived classes that will inherit convert. However, the function that will actually perform the conversion, compute (), is a pure virtual function that must be defined by the classes derived from convert. The specific nature of compute () will be determined by what type of conversion is taking place.
2. Implement CPP to demonstrate Exception Handling for Gmail Account Login OR ATM Pin Verification.
3. Write a C++ program to implement stack of characters and integers using function template.
4. Implement student Data Base Application with following functionality
 - a. Create and write student data in the file

Program: Computer Engineering

- b. Display File contents
- c. Search a record based on Roll no. and name
- d. Modify a record
- e. Delete record

Write the output in the output File

5. Write a C++ program to generate Country-Currency chart of all countries across the globe using MAP Container

Practice Practical Assignment (Optional) :

1. Write a program to find given number is prime or not.
2. C++ program to check entered character vowel or consonant.
3. Write C++ program to find the sum of first and last digit of any number.
4. Write C++ program to print number in words using switch case.
5. Write C++ program to find reverse of a number using recursion.
6. Given an array a[] of size N which contains elements from 0 to N-1, you need to find all the elements occurring more than once in the given array.
7. Given a string S Consisting Letters. Return the non-repeating character in S. If there is no non-repeating character, return '\$'.
8. Develop a class Counter that represents a simple integer counter. The class should satisfy the following requirements:
 - a. A constructor should be provided that takes a single int argument that is used to initialize the counter value. The argument should default to zero.
 - b. The prefix increment and postfix increment operators should be overloaded in order to provide a means by which to increment the counter value.
 - c. A member function get Value should be provided that returns the current counter value.In addition, the class must track how many Counter objects are currently in existence. A means for querying this count should be provided. The code must not use any global variables. (Hint: Use static members.)
9. Write a program that defines a shape class with a constructor that gives value to width and height. The define two sub-classes triangle and rectangle, that calculate the area of the shape area (). In the main, define two variables a triangle and a rectangle and then call the area() function in this two variables.
10. C++ Programs and Code Examples using Classes and Objects Calculate electricity charges.
11. Write C++ program to simulate file copy using file handling.
12. Develop a template function min3 that takes three arguments of the same type and returns the least of these arguments. For example, min3 (1, 0, 2) would return 0 and min3(1.5, 0.5, 3.0) would return 0.5.
13. Write a exception handling program to check the following conditions and throw the exception if the criterion does not met.
 - a. User has age between 18 and 55
 - b. User stays has income between Rs. 50,000 – Rs. 1,00,000 per month
 - c. User stays in Pune/ Mumbai/ Bangalore / Chennai
 - d. User has 4-wheelerAccept age, Income, City, Vehicle from the user and check for the conditions mentioned above. If any of the condition not met then throw the exception.
14. Write a program using STL stack to reverse given binary number.
15. Write a C++ program using map associative container for phone contact directory.

Program: Computer Engineering

Second Year Computer Engineering (2022 Course) Computer Organization and Operating System Laboratory			
Course Code	COPCC309	Credits	2
Contact Hours.	2 Hrs./Week (P)	Type of Course	Practical
Examination Scheme	Term Work 25 Marks		

Pre-requisites:

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External/ Internal	Marks
1.	Term work	Internal	25

Course Objectives	
1	To explain the structure, function and characteristics of computer systems.
2	To illustrate the structure and function of I/O and memory Organization.
3	To acquaint the basic of 8086 microprocessor and assembly language.
4	To make aware of different services of operating system and Linux commands

Course Outcomes : Students will be able to	
309.1	Demonstrate different hardware components of computer system.
309.2	Install operating systems and input output peripherals.
309.3	Implement 8086 assembly language program.
309.4	Implement process scheduling algorithm.
309.5	Make use of different Linux commands to perform various operations.

List of Experiments
Each student must perform all assignments
1. Assemble the computer system using various hardware components. Study the general structure of computer system.
2. Study of memory hierarchy with respect to capacity, access time, performance and cost per bit.
3. Single boot OS installation (Linux OS installation)
4. Dual boot OS installation (Linux / Windows OS installation)
5. Multi OS installation
6. Demonstrate installation of I/O Peripherals- Printer/Scanner
7. Study of the architecture of 8086 microprocessor

Program: Computer Engineering

- | |
|--|
| <p>8. Implement 8/16/32/64 bit ALP to accept two digit numbers and display same on command prompt</p> |
| <p>9. Implement 8/16/32/64 bit ALP to accept two digit numbers perform addition and display result on command prompt.</p> |
| <p>10. Write a program to implement Scheduling Algorithms: FCFS, SJF and Round Robin.</p> |
| <p>11. Study of various Linux commands.</p> |

Second Year Computer Engineering (2022 Course)			
Vedic Mathematics			
Course Code:	COHSM310	Credit	1
Contact Hours:	1 Hrs/week (L)	Type of Course:	Lecture
Examination Scheme	TW Evaluation 25 Marks		

Pre-requisites: Vedic Sutras, Vedic Sub Sutras

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External/ Internal	Marks
1.	Term work Evaluation	Internal	25

Course Objectives

1	To develop the understanding of techniques/ sutras to solve mathematical arithmetic's in easy and faster way and use these techniques in various competitive examinations.
2	To improve speed and efficiency to solve even the most complex mathematical problems.
3	To remove the phobia about mathematics in the minds of students.
4	To help students to have better command over mathematical concepts and boost up their self-confidence level towards the subject.

Course Outcomes: Students will be able to

310.1	Apply Vedic Mathematics techniques to Perform quickly and accurately mathematical calculations like multiplication, division, squares, cubes, LCM, HCF.
310.2	Apply Vedic Mathematics techniques to solve Linear Equations, Quadratic Equations, Factorization of a Cubic Polynomial.
310.3	Apply Vedic Mathematics techniques to Perform calculations in Coordinate Geometry, Differentiation, Integration and Trigonometry without relying heavily on calculators or written methods.

Topics covered:

UNIT I: Basic level (4 hrs.)
Introduction of Vedic Mathematics, Multiplication, Square, Cube, Divisibility Test , Highest Common Factor of Polynomials, Multiplication of Polynomials, Division of Polynomials.
UNIT II: Intermediate Level (4hrs.)
Linear Equations, Quadratic Equations, Factorization of a Cubic Polynomial, Magic squares, Dates and Calendar.
UNIT III: Advance Level (3 hrs.)
Determinant, Coordinate Geometry, Differentiation, Integration, Trigonometry.

Syllabus contents required for competitive exams (GATE, UPSC, MPSC etc.)(if complete unit is applicable then write only "unit 1/2/.." or write the contents from that unit):1. NA.

Textbooks

1. Advanced Vedic Mathematics, Rajesh Kumar Thakur.
2. Vedic Mathematics Made Easy, Dhaval Bathia
3. VEDIC MATHEMATICS For Students: LEVEL – 1 OF 5 SERIES, by Nava Vision

Reference books

1. Sri Bharatikrishna Tirthaji, "Vedic Mathematics", Published by Motilal Banarsidass, 1965. ISBN 81-208-0163-6.
2. Williams K.R. "Discover Vedic Mathematics" Vedic Mathematics Research Group, 1984. ISBN 1-869932-01-3.
3. Williams K.R. and M.Gaskell "The Cosmic Calculator". Motilal Banarsidass, 2002. ISBN 81-208-1871-7.
4. Nicholas A.P., Williams, J. Pickles. "Vertically and Crosswise". Inspiration books, 1984. ISBN 1-902517-03-2.

Program: Computer Engineering

Second Year Computer Engineering (2022 Course) Design Thinking			
Course Code	COHSM401	Credits	2
Contact Hours.	2 Hrs./Week (L)	Type of Course	Lecture/Tutorial
Examination Scheme	TW : 50 Marks	ORAL: -- Marks	

Pre-requisites:

Course assessment methods / tools:

Sr.No.	Course assessment methods/tools	External/ Internal	Marks
1.	TERM-WORK	Internal	50

Course Objectives	
1	To familiarize students with design thinking concepts and principles
2	Apply design thinking phases to create successful prototypes

Course Outcomes: Students will be able to	
401.1	Apply design thinking approaches to prepare design for identified problems.
401.2	Evaluate different ideation methods and demonstrate different feasible methods for designing problem statement.
401.3	Implementation of identified problems
401.4	Test the prototype to evaluate a design

Topics covered:	
UNIT I: BASICS OF DESIGN THINKING	(03 Hours)
Introduction: Definition of Design thinking, Basics of Design thinking, Design thinking principles – human centric rule, ambiguity rule, redesign rule, tangibility rule	
UNIT II: DESIGN THINKING PROCESS	(03 Hours)
Design thinking process – empathize- Empathize with the users -steps in empathize, define, ideate, prototype & test	
UNIT III: IDEATE PHASE	(03 Hours)
Customer centricity, What is ideation, Need for ideation, Uses of ideation, Ideation methods, Brainstorming, Rules for brainstorming, Mind maps ,Guidelines to create mind maps.	
UNIT IV: PROTOTYPE IMPLEMENTATION & TESTING	(03 Hours)
What is Prototype? Why Prototype? Rapid Prototype Development process, Guidelines for proto type , need of testing, testing prototypes, guidelines for planning a test.	

Program: Computer Engineering

Mini Project:-

Design Mini Project for any real world application by using following techniques of design thinking & scenario

- 1) Problem identification & definition
- 2) Customer requirements.
- 3) Brainstorming Sessions
- 4) Mind Map
- 5) Creating Prototype
- 6) Testing Prototype.

Scenario 1: -

GE Healthcare is an example of a company that focused on user-centricity to improve a product that seemingly had no problems. Healthcare launched the “Adventure Series.” This redesign initiative focused on making magnetic resonance imaging (MRI) machines more child-friendly.

GE Healthcare was able to craft a creative solution that was not only fun but increased patient satisfaction scores by 90 percent. This also yielded unexpected successes, including improved scan quality of pediatric patients, and ultimately saved customers time and resources.

Scenario 2: -

Although many companies have successfully used design thinking, Netflix has repeatedly leveraged it to become an industry giant. When the company realized DVDs were becoming outdated, it created an on-demand streaming service to stay ahead of the curve. This also inadvertently eliminated the inconvenience of having to wait for DVDs.

Netflix took its design thinking one step further and responded to customers’ need for original, provocative content that wasn’t airing on traditional networks. It improved its user experience by adding short trailers to its interface. Each of Netflix’s major updates was in response to customers’ needs and driven by an effective design thinking process.

Scenario 3: -

A prime example of this is UberEats’s Walkabout Program, where designers observe cities in which the company operates. Some elements they inspect are food culture, cuisine, infrastructure, delivery processes, and transportation. One of the innovations that came from their immersive research is the driver app, which focuses on delivery partners’ pain points around parking in highly populated urban areas. To address this, the driver app provides step-by-step directions from restaurant to customer to ensure smoother delivery processes. Understanding that pain points vary between geographic locations helps UberEats implement effective upgrades to its service that solve problems in specific locations

Text Books

1. Design Thinking , Gavin Ambrose, Paul Harris, AVA Publishing Deitel, “C++ How to Program”, 4th Edition, Pearson Education, ISBN:81-297-0276-2
2. Handbook of Design Thinking - Tips & Tools for how to design thinking, Christian Mueller- Rotenberg
Change by Design: How Design Thinking Transforms Organizations and Inspires

Program: Computer Engineering

Innovation by Tim Brown

Reference Books:

1. Design Thinking for Strategic Innovation: What They Can't Teach You at Business or Design School, Idris Mootee, Wiley Designing for Growth: a design thinking tool kit for managers, Jeanne Liedtka and Tim Ogilvie

MOOC/ Video Lectures available at:

1. <https://youtu.be/4nTh3AP6knM>

Program: Computer Engineering

Second Year Computer Engineering (2022 Course) Applied Mathematics and Computational Statistics			
Course Code:	COBSC402	Credit	3
Contact Hours:	3Hrs/week (P)	Type of Course:	Lecture
Examination Scheme	In-sem. Evaluation 40 Marks	End-sem. Examination 60 Marks	

Pre-requisites: Engineering Mathematics I, Engineering Mathematics II and Discrete Mathematics

Course assessment methods/tools:

Sr.No.	Course assessment methods/tools	External/Internal	Marks
1.	In-Sem. Evaluation	Internal	40
2.	End Semester Examination	External	60

Course Objectives	
1	To explain the concept of linear differential equations of higher order and its applications.
2	To impart the knowledge of Transform techniques such as Fourier transform, Z-transform
3	To develop the problem solving skill using Statistical analysis and Probability theory.
4	To explain the concept of Differential operator and Directional derivatives.

Course Outcomes: Students will be able to	
402.1	Solve the higher order linear differential equations and apply it to simulations, machine learning algorithms and data mining.
402.2	Find the Fourier and Z-Transform and apply it to solve Difference equations and conversion of time domain into frequency domain.
402.3	Analyze Statistical data using descriptive statistical techniques and use it in data science.
402.4	Apply probability theory and hypothesis tests to predict and analyze the data.
402.5	Interpret Statistical data using standard procedures of statistical modeling.
402.6	Find the directional derivatives of vector function and apply it to optimize Machine Learning algorithms.

Topics covered:
UNIT I: LINEAR DIFFERENTIAL EQUATIONS (LDE) (6hrs.) Introduction of Linear and Nonlinear differential equations, linear differential equation of nth order with constant coefficients, General method, Shortcut methods, Method of Variation of Parameters, Applications of Linear differential equations in machine learning algorithms and ANN.

Program: Computer Engineering

UNIT II: TRANSFORMS

(6hrs.)

Fourier Transform (FT): Complex exponential form of Fourier series, Fourier integral theorem, Fourier Sine & Cosine integrals, Fourier transform, Fourier Sine and Cosine transforms and their inverses. Applications to MP3 Encoding/ Sound Processing, transform time domain into frequency domain, image processing.

Z - Transform (ZT): Introduction, Standard properties, Z Transform of discrete functions and their inverses. Solution of difference equations. Applications to analysis of discrete-time systems digital signal processing, control theory.

UNIT III: STATISTICS

(6hrs.)

Measures of central tendency, Measures of dispersion, Coefficient of variation, Moments, Skewness and Kurtosis, Measures of Variability: Standard Deviation, Variance, Quartiles, Interquartile Range. Curve fitting: fitting of straight line, parabola and related curves, Correlation and Regression, Reliability of Regression Estimates, Applications to Machine learning and data science.

UNIT IV: PROBABILITY AND PROBABILITY DISTRIBUTION

(6hrs.)

Probability, Theorems on Probability, Bayes theorem, Random variables, Mathematical Expectation, Probability density function, Probability distributions: Binomial, Poisson, Normal and Hyper geometric, Applications to machine learning, artificial intelligence, natural language processing, and cryptography

UNIT V: SAMPLING & INFERENCE

(6hrs.)

Sampling: Introduction, Types of Sampling, Statistical Significance (p Values), Hypothesis testing, types of errors, level of significance, Pearson's test, z-test, chi square test, ANOVA test, Applications to Machine Learning and data science

UNIT VI: VECTOR DIFFERENTIATION

(6hrs.)

Introduction, Vector differential operators, Gradient, Divergent, Curl, Physical Interpretation of Vector Differentiation, Directional Derivatives, Solenoidal, Irrotational and conservative fields, Scalar Potential, Applications to Deep optimization technique of Machine Learning algorithms.

Syllabus contents required for competitive exams

Text Books:

1. B.V. Ramana, "Higher Engineering Mathematics", Tata McGraw-Hill, 34 edition, ISBN 13:9780070634190
2. Erwin Kreyszig, "Advanced Engineering Mathematics" Wiley Eastern Ltd., 10 Edition, ISBN 13: 9780470458365
3. Higher Engineering Mathematics by B. S. Grewal (Khanna Publication, Delhi).

Reference Books:

1. Advanced Engineering Mathematics, 2e, by M. D. Greenberg (Pearson Education)
2. Advanced Engineering Mathematics, 7e, by Peter V. O'Neil (Cengage Learning)
3. Differential Equations, 3e by S. L. Ross (Wiley India)
4. Introduction to Probability and Statistics for Engineers and Scientists, 5e, by Sheldon M. Ross (Elsevier Academic Press)

MOOC/ Video Lectures available at:

1. NPTEL Course lectures links: <https://nptel.ac.in/courses/111/105/111105090/> (Probability)
2. https://onlinecourses.nptel.ac.in/noc20_ma13/ (Advanced Engineering Mathematics)
V-lab (IIT-Bombay) link: http://vlabs.iitb.ac.in/vlabs-dev/labs/numerical_lab/labs/explist.php

**Second Year Computer Engineering (2022Course)
Advanced Data Structures and Algorithm**

Course Code:	COPCC403	Credit	3
ContactHours:	3 Hrs/week (L)	Type of Course:	Lecture
Examination Scheme	In-sem. Evaluation 40 Marks	End-sem. Examination 60 Marks	

Pre-requisites:

- **Programming and Problem Solving-I,II**

Course assessment methods/tools:

Sr.No.	Course assessment methods/tools	External/Internal	Marks
1.	In-Sem. Evaluation	Internal	40
2.	End Semester Examination	External	60

Course Objectives

1	To understand various algorithmic strategies to approach the problem solution.
2	To learn and understand representation, implementation and applications of trees, search trees, graphs, multiway trees and Hashing data structures.
3	To suggest appropriate data structures for solving computational problems.
4	To prepare students for solving competitive programming problems.

Course Outcomes: Students will be able to

403.1	Explain the fundamental analysis and Identify appropriate algorithmic strategy for solving problem.
403.2	Illustrate collision resolution techniques and implement it.
403.3	Describe and implement operations and applications of trees data structure.
403.4	Solve problems of Search trees and Heap.
403.5	Make use of Graph Data structures to implement real world applications.
403.6	Describe pattern matching algorithms and trie trees.

Topics covered:

UNIT I: ANALYSIS OF ALGORITHMS AND ALGORITHMIC STRATEGIES (5hrs.)

Analysis: Input size, best case, worst case, average case. Divide and conquer strategy-Principle, control abstraction, time analysis of control abstraction, Binary search

Greedy strategy: Principle, control abstraction, time analysis of control abstraction, knapsack problem,

Dynamic Programming: Principle, control abstraction, time analysis of control abstraction. 0/1 knapsack

Backtracking: Principle, control abstraction, time analysis of control abstraction, 8-queen problem

UNIT II: HASHING (6hrs.)

Hashing: General idea of Hashing, Hash Table, Hash function, Rehashing, Issues in Hashing, Collision Resolution Strategies: Linear Probing, Quadratic Probing, Double Hashing, Open addressing and Chaining.

Case study: Dictionary, Telephone directory

UNIT III: TREES (8hrs.)

Trees- basic terminology, General tree and its representation, representation using sequential and linked organization, Binary tree- properties, converting tree to binary tree, binary tree traversals(recursive and non-recursive)- inorder, preorder, post order, depth first and breadth first, Operations on binary tree. Binary Search Tree (BST), BST operations.

Case Study: Huffman's coding

UNIT IV: SEARCH TREES AND HEAP (6hrs.)

Search trees: Height Balanced Tree- AVL tree. Red-Black Tree, AA tree

Heap basic concepts, Realization of Heap, Heap as an Abstract Data Type, Heap implementation, Heap Sort, Heap as a Priority Queue.

Case Study: B,B+ tree

UNIT V: GRAPHS (6hrs.)

Graphs: Basic Terminologies, Storage Representation, Graph Traversals, Graph as Abstract Data Type, Spanning Trees, Minimum Spanning Trees, Kruskal's Algorithm, Prim's Algorithm, Dijkstra's Single Source Shortest Path Algorithm, Topological Sorting

Case Study : Google map

UNIT VI: PATTERN MATCHING AND TRIES (5hrs.)

String Matching: Introduction, Naïve string matching algorithm, Brute force, Robin – Karp algorithm, Boyer –Moore algorithm, Knuth – Morris – Pratt algorithm , Standard Tries, Compressed Tries, Suffix tries.

Syllabus contents required for competitive exams (GATE, UPSC, MPSC etc)

All Units are required for GATE exam

Text Books:

1. Horowitz Sahani, Dinesh Mehata, "Fundamentals of Data Structures in C++", Galgotia Publisher, ISBN: 8175152788, 9788175152786.
2. MFolk, BZoellick, G.Riccardi, "File Structures", Pearson Education, ISBN: 81-7758-37-5
3. Peter Brass, "Advanced Data Structures", Cambridge University Press, ISBN: 978-1-107-43982-5

Reference Books:

1. A. Aho, J. Hopcroft, J. Ulman, "Data Structures and Algorithms", Pearson Education, ISBN-0-201-43578-0.
2. Michael J Folk, "File Structures an Object Oriented Approach with C++", Pearson Education, ISBN: 81-7758-373-5

MOOC/ Video Lectures available at:

Data Structures and Algorithms: Coordinated by IIT, Delhi:

<https://nptel.ac.in/courses/106/102/106102064/>

Second Year Computer Engineering (Autonomous Course)
Data Communication and Computer Network

Course Code:	COPCC404	Credit	3
Contact Hours:	3 Hrs/week (L)	Type of Course:	Lecture
Examination Scheme	In-Sem 40 marks	End-Sem 60 marks	

Pre-requisites: Computer Organization & Operating System

-

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External/ Internal	Marks
1.	In-Sem	Internal	40
2.	End-Sem	Internal	60

Course Objectives

1	To explain the basics of Computer Network
2	To identify different techniques for framing, error control, flow control and routing.
3	To describe role of protocols at various layers in the protocol stacks.
4	To develop network programming

Course Outcomes : Students will be able to

404.1	Explain different line coding and multiplexing techniques for different types of computer networks
404.2	Explain various flow control and error control protocols
404.3	Select different types of IP address to configure a network and determine various routing protocols
404.4	Describe difference between TCP and UDP protocols and implement TCP/UDP socket programs
404.5	Illustrate role of application layer with its protocols
404.6	Differentiate between various wireless standards

Topics covered:

UNIT-I: BASICS OF COMPUTER NETWORKS (6 hrs.)

Introduction, Communication System, Baseband, Broadband and Carrier Communication, transmission modes, Baud rate, bit rate, SNR, Channel Bandwidth and rate of communication.

Line Coding Techniques: Bipolar, Unipolar, RZ, NRZ, Manchester, Multiplexing techniques: TDM, FDM, WDM, and CDMA, TD-SCDMA, Types of networks

Case Study: To demonstrate different networking tools

UNIT-II: DATA LINK LAYER (6 hrs.)

Data Link Layer :Introduction, functions. Design Issues: Services to Network Layer, Framing.ARQ strategies: Error detection and correction, Parity Bits, Hamming Codes (11/12-bits) and CRC. Flow Control Protocols: Unrestricted Simplex, Stop and Wait, Sliding Window Protocol. WAN Connectivity: PPP and HDLC. MAC Sub layer: Multiple Access Protocols: Pure and Slotted ALOHA, CSMA, WDMA, CSMA/CD, CSMA/CA

Case Study: Various IEEE standards for types of networks

UNIT-III: NETWORK LAYER (6 hrs.)

Network Layer: Introduction: Functions of Network layer. Basics of IPV4, ICMPV4, IPV6 in detail Motivation, Features, Address representation, Unicast and Multicast addresses, Header format, ARP, RARP, Distance vector and link state routing algorithms, Routing protocols:RIP, OSPF, BGP

Case Study : How sub-networks in campus is done

UNIT-IV: TRANSPORT LAYER (6 hrs.)

Transport Layer: Process to Process Delivery, Services, TCP, UDP, Socket Programming, TCP Flow control, TCP congestion control, TCP in wireless network, Real time transport protocol, Stream control transmission protocol (SCTP), Quality of services (QoS), Differentiated services, Integrated services, socket programming

Case Study : Role of various transport protocols in television broadcast

UNIT-V: APPLICATION LAYER (6 hrs.)

Application Layer:

OSI Model Block Diagram, Application layer protocols: Functionality and header formats, HTTP, DNS, FTP, SMTP, TELNET, DHCP

Case Study: Study various FTP servers

UNIT-VI: WIRELESS NETWORKS (5 hrs.)

Basic Concepts in Wireless LAN:

Wireless Standards (802.11 a/b/g/n/ac/ad), Wireless LAN and Technology,Wireless application protocols(Architecture and application) WLAN standards: WiMax, ZigBee,Bluetooth

Case Study: Study of various VoIP applications like Skype, whatsapp, facebook messenger

Text books

1. Andrew S. Tenenbaum, "Computer Networks", PHI, ISBN 81-203-2175-8.
2. Kurose, Ross "Computer Networking a Top Down Approach Featuring the Internet", Pearson; ISBN-10: 0132856204,ISBN-13: 978-0132856201 (pdf available)
3. Georgios I. Papadimitriou, Andreas S. Pomportsis, P. Nicopolitidis, Mohammed S. Obaidat,"Wireless Networks", Wiley Student Edition ISBN: 978-0-470-84529-5

Reference Books:

1. Fourauzan B., "Data Communications and Networking", 4th edition, Tata McGraw Hill,Publications, 2006, ISBN 0 - 07 – 0634145
2. Olifer & Olifer ,”Computer Networks-principles, technologies & protocols for network design”, WILEY
3. Rajiv Ramaswami, Kumar Shivarajan, Glan Shasaki, “Optical Networks a PracticalPerspective”, Elsevier-Morgan Kaufmann ISBN: 978-0-12-374092-2 pdf
4. Vijay K, Garg, “Wireless Communication and Networking”, Elsevier Morgan Koffman, ISBN: 978-81-312-1889-1

MOOC/ Video Lectures available at:

- <https://www.mooc-list.com/tags/computer-networking>
- <https://www.coursera.org/courses?query=computer%20network>
- https://onlinecourses.nptel.ac.in/noc23_cs48/preview

**Second Year Computer Engineering (2022 Course)
Software Engineering and Modeling**

Course Code:	COPCC405	Credit	3
Contact Hours:	3 Hrs/week (L)	Type of Course:	Lecture
Examination Scheme	In-sem. Evaluation 40 Marks	End-sem. Examination 60 Marks	

Pre-requisites: PPS

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External/ Internal	Marks
1.	In-sem. Evaluation	Internal	40
2.	End-sem. Examination	external	60

Course Objectives: The student will have ability to:

1	Get the knowledge of basic software engineering methods and practices.
2	Define software requirements and requirement engineering.
3	Apply approaches for various design and their principle.
4	Explore testing in various domains.
5	Development of significant teamwork and project based experience

Course Outcomes : Students will be able to

405.1	Compare various software process models and identify where these models are applicable.
405.2	Prepare software requirement specifications for any software.
405.3	Design software systems using UML
405.4	Analyze software metrics and apply the concepts of software estimation models
405.5	Identify and apply various software testing strategies.

Topics covered:

UNIT I: INTRODUCTION TO SOFTWARE ENGINEERING (6 hrs.)

Nature of Software: Software Engineering, Software Process, Process models- Waterfall model, Incremental models, Evolutionary Models, Spiral Model, V-model, Agile Process model, Extreme programming(XP) Practices, Scrum model, RAD model.

Case Study:

<https://aktiasolutions.com/kanban-case-study/>

<https://uxplanet.org/case-study-defining-how-jira-operates-within-the-constraints-of-the-communication-tool-slack-46e036be272a>

UNIT II: SOFTWARE REQUIREMENTS ANALYSIS (6 hrs.)

Functional and Non-functional Requirements: Requirement Sources and Elicitation Techniques, Analysis Modeling for Function-oriented and Object-oriented Software Development, System and Software Requirement Specifications in IEEE Format, Requirement Validation, Traceability.

Case Study:

https://www.academia.edu/24074814/Library_Management_System_SRS_Report_Library_Management_System

Program: Computer Engineering

UNIT III: DESIGN ENGINEERING (6 hrs.)

The Software Design Process: Design Concepts and Principles, **Architectural Design:** Design Decisions, Views, Patterns, and Application Architectures. **Modeling Component level Design:** Component, Designing class-based components, Conducting Component-Level Design. **User Interface Design:** The Golden Rules, Interface Design Steps and Analysis.

Case Study: <https://blog.tubikstudio.com/case-study-tracking-app/>

UNIT IV: SOFTWARE MODELING (6 hrs.)

Software Modeling: What is Modeling, Importance of Modeling, Purpose of Modeling, Three models: 1.Class Model 2.State model 3. Interaction Model

Introduction to the UML Language:

Structural diagrams: Class diagram, Object diagram, Component diagram, Deployment diagram. Behavioral diagrams: Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram.

Case Study: <https://www.uml-diagrams.org/use-case-diagrams-examples.html/>

UNIT V: SOFTWARE METRICS (6 hrs.)

Software Metrics: Introduction to Software Metrics, Size-oriented metrics and function point metrics. Effort and cost estimation techniques -LOC-based and Function-point based measures - The COCOMO model, Putnam model, Function Point Analysis Based Methods.

Case Study:

https://www.academia.edu/6816948/Case_Study_on_Software_Cost_Estimation_Techniques

UNIT VI: Introduction Software Testing and Quality Management (6 hrs.)

Introduction to bugs, faults and failures and error **Testing Tactics:** White box and Black box tests, Validation and Verification Testing **Quality Management:** Quality Concepts, Quality Assurance.

Requirement Traceability matrix: Traceability Matrix (TM), Requirement Traceability Matrix, importance of RTM, Parameters to include in Requirement Traceability Matrix, How to Create RTM ,Comparison of Functional Requirement and Requirement Traceability matrix.

Case Study: <https://www.guru99.com/test-case.html>

https://www.softwaretestinghelp.com/sample-test-cases-testing-web-desktop-applications/#Checklist_1_Mobile_Testing_Checklist

Syllabus contents required for competitive exams (GATE, UPSC, MPSC etc.): Unit 1 and Software Testing and Quality Management

Text Books:

1. Roger s pressman, “software engineering: a practitioners approach”, McGrawHill, ISBN-0-07-337597-7
2. Ian Sommerville, “Software Engineering”l, Addison and Wesley, ISBN 0-13-703515-2

Reference Books:

1. Carlo Ghezzi, “Fundamentals of Software Engineering”, PHI, ISBN-10: 0133056996
2. Rajib Mall, “Fundamentals of Software Engineering”, PHI, ISBN-13: 978-8120348981
3. Pankaj Jalote, “An Integrated Approach to Software Engineering”, Springer, ISBN 13:9788173192715.
4. S K Chang, “Handbook of Software Engineering and Knowledge Engineering”, WorldScientific, Vol I, II, ISBN: 978-981-02-4973-1
5. Tom Halt, “Handbook of Software Engineering”, Clanye International ISBN-10: 1632402939
6. Brahma Dathan, Sarnath Ramnath, “Object-Oriented Analysis, Design, and Implementation”, Universities Press

MOOC/ Video Lectures available at:

https://swayam.gov.in/nd1_noc19_cs69/preview

https://swayam.gov.in/nd2_cec20_cs07/preview

Activity List:

Sr. No.	Activity based Assignments
1	Identify a software system and apply suitable process model.
2	Write Software requirement specification for the identified software system.
3	Represent Architectural, Component level and User Interface Design for the identified software system.
4.	Draw UML Diagrams for the identified software system.
5.	Estimate the Cost for the identified software system using COCOMO model, Putnam model, Function Point Analysis Based Methods.
6.	Write manual test cases for the identified software system.
7.	Students should be divided into sub-groups of 3to 4 students. The group of students completes the following tasks: Apply all software Engineering concepts and prepare documentation for the mini project.

Program: Computer Engineering

Second Year Computer Engineering (2022 Course) Programming in Java			
Course Code:	COVSE406	Credit	3
Contact Hours:	1 Hrs/week (L) 4 Hrs/week (PR)	Type of Course:	Lecture + Practical
Examination Scheme	TW 50 marks	PR 50 marks	

Pre-requisites: C++ Programming

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External/ Internal	Marks
1.	Term Work	Internal	50
2.	Practical	Internal	50

Course Objectives	
1	To develop an application with core Java concepts
2	To apply exception handling concepts in writing error free program.
3	To develop an application with advance Java concepts like Applet, Swing and AWT.

Course Outcomes : Students will be able to	
406.1	Make use of basic OOP concepts and control loops to write Java programs.
406.2	Implement a Java program using arrays and exception handling.
406.3	Apply concept of swing to develop application.

Topics covered:
UNIT-I: Introduction to Java and Control Structures (Lecture 3) Introduction to Java, Java's features and advantages, Setting up the Java development environment (IDE, JDK), Variables, data types, and declarations, Arithmetic operations and basic input/output, Conditional statements (if, else if, else), Switch statement, Loops (while, for).
UNIT-II: Java OOP concepts, Functions and Methods (Lecture 3) Classes and objects, Access Modifiers in Java, Encapsulation, inheritance, and polymorphism, Constructors, Copy constructor, Method overloading, Method Overriding Abstract Class, Abstract method, Interface in Java, Static Keyword, Final Keyword, Inner Class in Java, Super and this keyword.
UNIT-III: Arrays and Exception Handling (Lecture 3) Arrays and Collections: Declaring and using arrays, ArrayList and basic collection operations, Iterating through collections, type casting, Exception Handling, Handling exceptions using try-catch, Throwing exceptions and creating custom exceptions

Program: Computer Engineering

UNIT-IV: Building Interactive Desktop Applications (Lecture 3)

Overview of Swing and its importance in desktop applications, Building and running the first Swing application using JButton and ActionListener, JLabel and JTextField. Overview of JDBC and its role in Java applications, CRUD operation.

Text Books:

- Programming with Java A Primer, E. Balaguruswamy Tata McGraw Hill Companies

Reference Books:

- Java: The Complete Reference Hebert Schildt, Mc Graw Hill
- Object-Oriented Programming with C++ and Java Debasis Samanta, Prentice Hall India.

MOOC/ Video Lectures available at: https://onlinecourses.nptel.ac.in/noc22_cs102/preview

List of Practicals: Complete any 12

Group A : (Any 3)

1. Write a program in Java to generate first n prime numbers.
2. Write a java Program to generate a Ladder of number.
3. Write a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions?
4. Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome?
5. Write a Java Program to finds the average of numbers in an array.

Group B : (Any 7)

6. Implement Student class using following Concepts
 - a. All types of Constructors
 - b. Static variables and instance variables
 - c. Static blocks and instance blocks
 - d. Static methods and instance methods
7. Create a class with overloaded methods for performing mathematical operations (addition, subtraction, multiplication, and division) for both integers and doubles. Demonstrate these overloaded methods with different data types.
8. Write Java program to calculate area of triangle, square & circle using function overloading. Function parameter accept from user (Use function Overloading concepts and Inheritance).
9. Design an abstract class "Shape" with abstract methods for calculating area and perimeter. Create subclasses for specific shapes (e.g., Circle, Rectangle, Triangle) that extend the abstract class and implement these methods.
10. Define an interface "Employee" with methods for calculating salary and displaying employee information. Create classes that implement this interface for different types of employees (e.g., FullTimeEmployee, PartTimeEmployee).
11. Write a program that calculates the result of dividing two numbers. Implement exception handling to handle division by zero and display an appropriate error message./ Design a program that reads data from a file and handles exceptions, such as file not found or permission issued.
12. Write a program for following exception, develop a suitable scenario in which the following exceptions occur:
 - a. divide by zero
 - b. Array index out of bounds exception
 - c. Null pointer Exception
13. Write a program in Java to demonstrate the use of 'final' keyword in the field declaration. How it is accessed using the objects.

Program: Computer Engineering

14. Build a class hierarchy for bank accounts (e.g., Savings Account, Checking Account) with a common base class "Bank Account." Override the calculate Interest() method in the derived classes to calculate interest based on specific account types.
15. Design a Java program for a library catalog system. Create overloaded methods for adding and searching for books in the library. Overload the methods to accept various search criteria (e.g., title, author, ISBN) and demonstrate their usage in managing the library catalog.

Group C: (any 2)

16. Write an small application in Java to develop Banking Application in which user deposits the amount Rs 1000.00 and then start withdrawing of Rs 400.00, Rs 300.00 . Develop a GUI in java using Swing
17. Implement insert and display operations using JDBC Connectivity
18. Implement CRUD operations using JDBC Connectivity
19. Design a user input form using Swing components. Implement an Action Listener to display user input on button click.
20. Develop a basic calculator application using Swing.

Practice Practical Assignments:

1. Write a java program to find the Fibonacci series using recursive and non recursive functions.
2. Write a java program to find whether the give number is Armstrong number or not.
3. Write a java program to reverse the given string.
4. Develop a class Bank Account that implements the Comparable interface. Order an array list of bank accounts by increasing balance. Provide methods to determine the lowest and highest balance in the Array List.
5. There is a class Adder which has two data members of type 1D int array and int variable. It has two functions: getdata and num sum. Function getdata accepts non-empty array of distinct integers from user in 1D int array data member and a targetsum in another data member. The function numsum adds any two elements from an input array which is equal to targetsum and return an array of resulting two elements, in any order. If no two numbers sum up to the target sum, the function should return an empty array. Note that the target sum is to be obtained by summing two different integers in the array; you can't add a single integer to itself in order to obtain the target sum. You can assume that there will be at most one pair of numbers summing up to the target sum. Use constructor. Use extra variables if needed

Input:

Array=[3,5,-4,8,11,1,-1,7] targetsum=15

Output: [8,7]

Input:

Array=[3,5,-4,8,11,1,-1,6] targetsum=15

Output: []

6. Describe abstract class called Shape which has three subclasses say Triangle, Rectangle, Circle. Define one method area() in the abstract class and override this area() in these three subclasses to calculate for specific object i.e. area() of Triangle subclass should calculate area of triangle etc. Same for Rectangle and Circle
7. Write a java program in which you will declare an abstract class Vehicle inherits this class from two classes car and truck using the method engine in both display "car has petrol engine" and "truck has diesel engine".
8. Write a java program that works as a simple calculator. Use a Grid Layout to arrange Buttons for digits and for the + - * % operations. Add a text field to display the result.

Second Year Computer Engineering (2022Course) Advanced Data Structures and Algorithm Laboratory			
Course Code:	COPCC407	Credit	2
Contact Hours:	4Hrs/week (P)	Type of Course:	Practical
Examination Scheme	Term Work(TW) 25 Marks	Practical(PR)50 Marks	

Pre-requisites:

- **Programming and Problem Solving-I,II**

Course assessment methods/tools:

Sr.No.	Course assessment methods/tools	External/Internal	Marks
1.	Term Work	Internal	25
2.	Practical	External	50

Course Objectives

1	To understand various algorithmic strategies to approach the problem solution.
2	To learn and understand representation, implementation and applications of trees, search trees, graphs, multiway trees and Hashing data structures.
3	To suggest appropriate data structures for solving computational problems.
4	To prepare students for solving competitive programming problems.

Course Outcomes: Students will be able to

407.1	Identify appropriate algorithmic strategy for solving problem and analyze using asymptotic notations.
407.2	Illustrate collision resolution techniques and implement it.
407.3	Implement programs using non linear data structures like Trees, Graph, Search trees and Heap.
407.4	Develop logic for solving competitive programming problems.

List of Experiments:

1.	Write C++ code for merge sort and analyze time complexity.
2.	Implement 0/1 Knapsack problem or 8 Queen's problem
3.	Implement Cuckoo Hashing
4.	Implement any real time application with the help of hashing concept and handle collision using linear probing without replacement.
5.	A Dictionary stores keywords & its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/ Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Binary Search Tree for implementation.

Program: Computer Engineering

6. For given expression eg. $a-b*c-d/e+f$ construct inorder sequence and traverse it using post-order traversal(non recursive).
7. Shortest path algorithm implementation using Graph
8. Write a program to find all the Hamiltonian cycles in a connected undirected graph $G(V,E)$ using backtracking
9. Single source shortest path algorithm – Connected Components and finding a cycle in a graph
10. Implementation Max or Min Heap.
11. Implement word cloud program.

Practice Problems:

1. Solve following problem using 0/1 Knapsack problem

Consider the problem having weights and profits are:

Weights: {3, 4, 6, 5} Profits: {2, 3, 1, 4}

The weight of the knapsack is 8 kg The number of items is 4

2. Solve 4 Queens problem and draw state space tree.
3. Suppose you are given the following set of keys to insert into a hash table that holds exactly 11 values: 113 , 117 , 97 , 100 , 114 , 108 , 116 , 105 , 99 Which of the following best demonstrates the contents of the hash table after all the keys have been inserted using linear probing?
4. Solve double hashing collision resolution technique with any suitable example.
5. Solve quadratic probing hashing collision resolution technique with any suitable example.
6. Find ancestors of a given node in a binary tree.
7. What is binary tree? Construct binary tree from the given traversals.

Pre-order: G,B,Q,A,C,K,F,P,D,E,R,H

In-order: Q,B,K,C,F,A,G,P,E,D,H,R

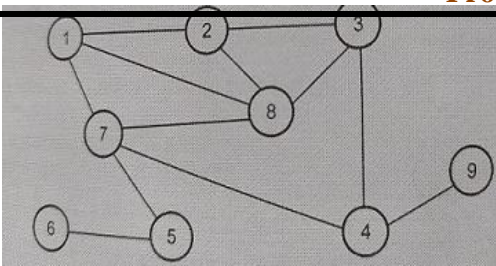
8. Construct binary search tree for the following data.
10,08,15,12,13,07,09,17,20,18,04,05
9. Construct binary expression tree from following expression:

$A+B*C*E+F$

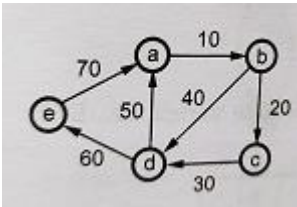
10. For the binary tree represented as an array, construct threaded binary tree :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
A	B	C	D	E	F	G		H	I				J	K														L

11. Construct an AVL tree by inserting numbers from 1 to 8
12. Build B tree of order 4 for following data:
30,31,23,32,22,28,24,29,15,26,27,34,39,36
13. Define DFS & BFS for a graph. Show DFS & BFS for the following graph with starting vertex as 1.



14. Solve all pair shortest path problem
For following diagraph.



15. There are flight paths between cities. If there is a flight between city A and city B then there is an edge between the cities. The cost of the edge can be the time that flight take to reach city B from A, or the amount of fuel used for the journey. Represent this as a graph. The node can be represented by airport name or name of the city. Use adjacency list representation of the graph or use adjacency matrix representation of the graph. Check whether the graph is connected or not. Justify the storage representation used.
16. Write C++ Program to Perform Naive String Matching.
17. Write C++ program for KMP Algorithm for Pattern Searching

Second Year Computer Engineering (Autonomous Course)

Computer Network Laboratory

Course Code:	COPCC408	Credit	1
Contact Hours:	2 Hrs/week (PR)	Type of Course:	Practical
Examination Scheme	Practical 50marks		

Pre-requisites:Computer Organization & Operating System

Course assessment methods/tools:

Sr.No.	Course assessment methods/tools	External/ Internal	Marks
1.	Practical	Internal	50

Course Objectives

1	To describe various components of a local area network
2	To identify different techniques for framing, error control, flow control and routing.
3	To describe role of protocols at various layers in the protocol stacks.
4	To develop network programming

Course Outcomes: Students will be able to

408.1	Use packet analyzer tools like wireshark to understand working of various protocols
408.2	Design a local area network and test the connectivity between two nodes
408.3	Implement data link layer protocols
404.4	Compare routing algorithms distance vector and link state routing
404.5	Implement TCP/UDP socket program
404.6	Analyze the application layer protocol: DNS

Topics covered:

1. Setup a wired LAN using Layer 2 Switch. It includes preparation of cable, testing of cable using line tester, configuration machine using IP addresses, testing using PING utility
2. Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC.
3. Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode
4. Demonstrate the different types of topologies and types of transmission media by using a packet tracer tool
5. Write a program to implement link state /Distance vector routing protocol to find suitable path for transmission.
6. Use packet Tracer tool for configuration of 3 router network using one of the following protocol RIP/OSPF/BGP

Program:Computer Engineering

7. Write a program using TCP socket for wired network for following
- Say hello to each other
 - File transfer
 - Calculator

OR

Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines

8. Write a program for DNS lookup. Given an IP address as input, it should return URL and vice-versa

Textbooks

- Andrew S. Tenenbaum, "Computer Networks", PHI, ISBN 81-203-2175-8.
- Kurose, Ross "Computer Networking a Top Down Approach Featuring the Internet", Pearson; ISBN-10: 0132856204,ISBN-13: 978-0132856201 (pdf available)
- Georgios I. Papadimitriou, Andreas S. Pomportsis, P. Nicopolitidis, Mohammed S. Obaidat,"Wireless Networks", Wiley Student Edition ISBN: 978-0-470-84529-5

Reference Books:

- Fourauzan B., "Data Communications and Networking", 4th edition, Tata McGraw Hill,Publications, 2006, ISBN 0 - 07 – 0634145
- Olifer&Olifer ,"Computer Networks-principles, technologies & protocols for network design", WILEY
- Rajiv Ramaswami, Kumar Shivarajan, GnanShasaki, "Optical Networks a PracticalPerspective", Elsevier-Morgan Kaufmann ISBN: 978-0-12-374092-2 pdf
- Vijay K, Garg, "Wireless Communication and Networking", Elsevier Morgan Koffman, ISBN: 978-81-312-1889-1

Program: Computer Engineering

Second Year Computer Engineering (2022 Course) Project Development Laboratory -I			
Course Code:	COELC409	Credit	1
Contact Hours:	2 Hrs/week (PR)	Type of Course:	Practical
Examination Scheme	Oral 25 marks		

Pre-requisites: Knowledge of problem solving, C, C++ and Python.

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External/ Internal	Marks
1.	Oral	Internal	25

Course Objectives	
1	To develop critical thinking and problem solving ability by exploring solutions to realistic/social problems.
2	To Evaluate alternative approaches, and justify the use of selected tools and methods,
3	To engage students in rich and authentic learning experiences.
4	To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
5	To develop an ecosystem to promote entrepreneurship and research culture among the students

Course Outcomes : Students will be able to	
409.1	Identify the real life problems occur in societies.
409.2	Compare alternative approaches to solve identified problem.
409.3	Analyze and synthesize the identified problem from technological perspective.
409.4	Design the reliable solution to meet challenges.
409.5	Evaluate the solution based on the criteria specified.

Course Relevance:

Project development lab adapted Project-based learning approach. It provides opportunity to student to develop knowledge and skills through engaging in projects. It is "learning by doing approach."

PBL is a style of active learning and inquiry-based learning.

In Project development lab -I is expected student has to search societal problem, challenge, or problem to research and respond to solve.

To tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain.

Project based learning will also redefine the role of teacher as mentor in the learning process.

It requires regular mentoring by faculty throughout the semester for successful completion of the idea/project tasks selected by the students.

Program: Computer Engineering

Group Structure:

- There should be a team/group of 4-5 students.
- A mentor teacher assigned to individual groups.

Teacher's Role:

- Teacher is not the source of solutions; rather he will act as the facilitator and mentor.
- To guide how to apply the principles of problems solving, critical thinking and metacognitive skills to solve selected problem.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and help them to empower.

Assessment:

- The mentor is committed to assessing and evaluating student performance.
 - Progress of lab is monitored regularly on weekly basis. Weekly review of the work is necessary.
 - Group demonstrates their knowledge and skills
1. Individual assessment for each student (Understanding individual capacity, role and involvement in the project)
 2. Group assessment (roles defined, distribution of work, intra-team communication and togetherness)
 3. Documentation and presentation

Evaluation and Continuous Assessment:

- It is recommended that all activities should to be recorded regularly, regular assessment of work need to be done and proper documents need to be maintained at college end by both students as well as mentor (work book).
- Continuous Assessment Sheet (CAS) is to be maintained by all mentors.

Recommended parameters for assessment/evaluation and weightage:

1. Idea Inception and Awareness /Consideration of -Environment/ Social /Ethics/ Safety measures/Legal aspects/Sustainable Development Goals (10%)
2. Outcomes of lab/Problem Solving Skills/Solution provided/Final product (Individual assessment and team assessment) (40%)
3. Documentation (Gathering requirements, design and modeling, implementation/execution, use of technology and final report, other documents) (25%)
4. Demonstration (Presentation, User Interface, Usability) (20%)
5. Contest Participation/ publication/ Blog/ Git Hub upload (5%)

Selection of Project/Problem:

- Students must focus to initiate the task/idea. The idea inception and consideration shall be from following areas as a real world problem: Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- Also consider 17 sustainable development goals while selecting problems. These goals are 1. End poverty in all its forms everywhere, 2. Zero Hunger, 3. Ensure healthy lives and promote well-being for all at all ages, 4. Quality Education, 5. Achieve gender equality and empower all women and girls, 6. Ensure access to water and sanitation for all, 7. Ensure access to affordable, reliable, sustainable and modern energy, 8. Sustainable economic growth, employment and decent work for all, 9. Build resilient infrastructure, promote sustainable industrialization and foster innovation, 10. Reduce inequality within and among countries, 11. Make cities inclusive, safe, resilient and sustainable, 12. Ensure sustainable consumption and production patterns, 13. Combat climate change and its impacts, 14. Conserve and sustainably use the oceans, seas and marine resources, 15. Conserving life on land, 16. Promote just, peaceful and inclusive societies, 17. Revitalize the global partnership for sustainable development (Ref. <https://sdgs.un.org/goals>)

Program: Computer Engineering

- Sample list to start with Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or “wondering”. This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.

Student's Role:

Students must have the ability to initiate the task/idea.

Check feasibility and survey of existing solutions.

They must learn to think.

Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.

Students in this are actively constructing their knowledge and understanding of the situation in groups.

Students must work in team, develop communication skill and project and finance management skills.

Students are expected to work in groups.

They have to develop interpersonal and group process skills.

They must meet weekly to mentor and given review to mentor.

Skills developed:

Inquiry Skills

- Students are expected to develop critical thinking abilities by constantly relating:
 - Who benefits from this?
 - What are the problem/ challenge?
 - What are the strength and weaknesses?
 - What is another alternative solution?
 - What is best/worst case scenario?
 - Where is there most need for this?
 - Where we can get more information?
 - Where are the areas of improvement?
 - When this would benefit our society?
 - How does benefit us/others?

Information Literacy

Information literacy is an integral part of self- directed learning Information literacy involves the ability to:

- Know when there is a need for information
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.

Collaborative learning

- It is an educational approach to teaching and learning that involves groups of students working together to solve a problem or complete a project
- In collaborative learning, learners have the opportunity to talk with peers, exchange diverse beliefs present and defend ideas, as well as questioning other ideas.

Interpersonal Skills

- Interpersonal skills relating to group process are essential for effective problem solving and learning.
- It is important that students are made aware of these inter personal skills.
- Conflict management skills and Team leadership skills.

Program: Computer Engineering

- Students who have these skills have a better opportunity to learn than students who do not have these skills and Time Management

Meta-cognitive Skills

- Students need to reflect on the processes they are using during the learning process.
- Compare one strategy with another, and evaluate the effectiveness of the strategy used

Develop project using technologies C, C++ , Python, Java, HTML and IOT.

Sample Project Statement

- Design a system for Bus Management system.
- Design a system for Vehicle Parking Management System.
- Design a system for Food ordering system.
- Develop app like urban clap to provide services to societies.
- Design and deploy a system for real-time writing Devnagari character detection and conversion in Unicode format.
- Design a system for Airline reservation.
- Design a system for bike pooling or car pulling for students and staff.
- Design a app to calculate average of car.
- Develop app to locate hospitals and ambulance in emergency.
- Design a system to detect potholes and report to corporation

...not limited to.....Faculty and students are free to include other area which meets the society requirements at large.

Text Books:

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; (https://www.aishe.org/wp-content/uploads/2017/05/Full-Book-A-New-Model-Of-Problem-Based-Learning-Terry-Barrett_book.pdf)
2. Problem Based Learning. By Mahnaz moallem, woei hung and Nada Dabbagh,Wiley Publishers.(https://elearning.amu.edu.et/pluginfile.php/19747/mod_resource/content/1/PBL.pdf)

Reference Books:

HOTS-Oriented Module: Project-Based Learning by Hasanatul Hamidah Talitha Ardelia Syifa Rabbani Susi Fauziah Rizma Angga Puspita Reski Alam Gasalba Nirwansyah.(<https://repositori.kemdikbud.go.id/21381/1/Project-Based%20Learning.pdf>)
Gopalan,” Project management core text book”, Indian Edition
James Shore and Shane Warden, “The Art of Agile Development. (https://poetiosity.files.wordpress.com/2011/04/art_of_agile_development.pdf)

MOOC/ Video Lectures available at:

Problem Based learning: https://onlinecourses.swayam2.ac.in/ntr20_ed12/preview

Second Year Computer Engineering (2022 Course)
Audit course- Sustainable Development Goals

Course Code:	COHSM410	Credit	1
Contact Hours:	1 Hrs/week (P)	Type of Course:	Lecture
Examination Scheme	Term-work 25 Marks		

Pre-requisites: Nil

Course assessment methods/tools:

Sr. No.	Course assessment methods/tools	External/ Internal	Marks
1.	Term-work	Internal	25

Course Objectives

1	To discuss the sustainable development goals.
2	To explain framework of Seventeen Sustainable Development Goals.
3	To discuss structure and order of Sustainable Development Goals.
4	To study cases of Sustainable Development Goals.

Course Outcomes: Students will be able to

412.1	Explain sustainable development goals.
412.2	Describe framework of Seventeen Sustainable Development Goals.
412.3	Discuss structure and order of Sustainable Development Goals.
412.4	Report case studies of Sustainable Development Goals.

Topics Covered

Unit 1: Introduction to SDGs(3 hrs)

Sustainability, Sustainable development, Role of UN and the Need for SDGs, Scope and Inclusion and Agenda 2030, Our Common Future and Philosophy behind SDGs, Distinction between Development and Sustainable Development

Unit 2: Sustainable Development Goals (5 hrs)

Framework and Structuring of Seventeen SDGs

SDG 1: No Poverty

SDG 2: Zero Hunger

SDG 3: Good Health and Well-being

SDG 4: Quality Education

SDG 5: Gender Equality

SDG 6: Clean Water and Sanitation

SDG 7: Affordable and Clean Energy

SDG 8: Decent Work and Economic Growth

SDG 9: Industry, Innovation and Infrastructure

SDG 10: Reduced Inequality
SDG 11: Sustainable Cities and Communities
SDG 12: Responsible Consumption and Production
SDG 13: Climate Action
SDG 14: Life Below Water
SDG 15: Life on Land
SDG 16: Peace and Justice Strong Institutions
SDG 17: Partnerships to achieve the Goal

Unit 3: SDG Structure and Order (3 hrs)

Interrelationships and Connections between Seventeen SDGs, SDG Structure and Order at Levels of People, Ecological and Spiritual , SDGs and Socio Ecological Systems: Economy; Society; Biosphere.

Unit 4: Sustainable Development Goals- Case Studies (2 hrs)

Case Studies from around the World, Case studies from India

Text Books

1. Hazra, Somnath., Bhukta, Anindya (2020) Sustainable Development Goals An Indian Perspective, Springer International Publishing, Switzerland
2. Ziai, Aram (2016) Development Discourse and Global History from colonialism to the sustainable development goals. Routledge, London & New York
3. OECD (2019), Sustainable Results in Development: Using the SDGs for Shared Results and Impact, OECD Publishing, Paris, <https://doi.org/10.1787/368cf8b4-en>.
4. Sachs, J., Schmidt-Traub, G., Kroll, C., Lafortune, G., Fuller, G., Woelm, F. 2020. The Sustainable Development Goals and COVID-19. Sustainable Development Report 2020. Cambridge: Cambridge University Press.

Relevant websites, movies, and documentaries

<https://www.un.org/sustainabledevelopment/>