(An Autonomous College under VTU)

Department of Computer Science and Engineering

III to VIII Semester Scheme and Syllabus

Scheme – 2023

As per the NEP 2020 Guidelines, Choice-Based Credit System

&

Outcome-Based Education

w.e.f.

Academic Year 2024-2025

VISION

Create **globally competent professionals** through quality education in the field of Computer Science and Engineering.

MISSION

M1: Empowering students by imparting knowledge, latest technologies through practical approach and academic professionalism to fulfill the needs of the industry.

M2: Developing Technical proficiencies, communication skills and teamwork among the students.

M3: Inculcating ethics, social behavior, and universal human values for sustainable societal growth and environmental protection.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Apply the Computer Science and Engineering technical knowledge to fulfil the organizational needs.

PEO2: Inculcate proficiency, good communication skill, team building and problem-solving abilities.

PEO3: Pursue higher studies, research and to become entrepreneurs with good work ethics and professional behaviour.

PROGRAM OUTCOMES (POs)

Graduates of the Computer Science and Engineering Programme will be able to achieve the following POs:

PO1: Engineering Knowledge

Apply the knowledge of mathematics, science, engineering fundamentals, and Computer Science and Engineering principles to the solution of complex problems in Computer Science and Engineering.

PO2: Problem Analysis

Identify, formulate, research literature, and analyze complex Computer Science and Engineering problems reaching substantiated conclusions using first principles of mathematics and engineering sciences.

PO3: Design/Development of Solutions

Design solutions for complex Computer Science and 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.

PO4: Conduct investigations of Complex problems

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 related to Computer Science and Engineering problems.

PO5: Modern Tool Usage

Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex Computer Science and Engineering activities with an understanding of the limitations.

PO6: The Engineer and Society

Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional Computer Science and Engineering practice.

PO7: Environment and Sustainability

Understand the impact of the professional Computer Science and Engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics

Apply ethical principles and commit to professional ethics and responsibilities and norms of the Computer Science and Engineering practice.

PO9: Individual and Team work

Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication

Communicate effectively on complex Computer Science and 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.

PO11: Project Management and Finance

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 Computer Science and Engineering projects and in multidisciplinary environments.

PO12: Life Long Learning

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.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Professional Skills:

Ability to analyse, design and develop computer programs in the areas related to system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexity.

PSO2: Problem-Solving Skills:

Ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.

PSO3: Foundation of mathematical concepts:

Ability to apply mathematical concepts to solve real world problems using appropriate data structure and suitable algorithms.

NAGARJUNA COLLEGE OF ENGINEERING & TECHNOLOGY, BENGALURU

B.E. in Computer Science and Engineering
Scheme of Teaching and Examinations 2023
Outcome Based Education (OBE) and Choice Based Credit System (CBCS) (Effective from the academic year 2024-25)

III SEMESTER

Sl. No	Course	Course Code	Course Title	Teaching Department (TD) and Question Paper Setting (PSB)	Teaching Hours /Week				Examination				Credits
					Theory Lecture	Tutorial	Practical	SDA	Duration in Exams	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1	PCC/BSC	23MAT31	Mathematics for Computer Science	Maths Dept.	3	2	0		03	50	50	100	4
2	IPCC	23CSI32	Digital design & computer organisation	CSE	3	0	2		03	50	50	100	4
3	IPCC	23CSI33	Operating systems	CSE	3	0	2		03	50	50	100	4
4	PCC	23CST34	Data structures and applications	CSE	3	0	0		03	50	50	100	3
5	PCCL	23CSL35	Data structure Lab	CSE	0	0	2		03	50	50	100	1
6	ESC	23CSI36X	ESC/ETC/PLC	CSE	2	0	2		03	50	50	100	3
7	UHV	23UHV37	Social Connect and Responsibility	Any Department	0	0	2		01	100	---	100	1
8	AEC/ SEC	23CSL38X	Ability Enhancement Course/Skill Enhancement Course – III	CSE	If the Course is a Theory: 1	0	0		01 / 02	50	50	100	1
					If a Course is Laboratory: 0	0	2						

9	MC		National Service Scheme (NSS)	NSS coordinator	0	0	2			100	---	100	0
			Physical Education (PE) (Sports and Athletics)	Physical Education Director									
			Yoga	Yoga Teacher									
Total										**550**	**350**	**900**	**21**

PCC: Professional Core Course, **PCCL**: Professional Core Course laboratory, **UHV**: Universal Human Value Course, **MC**: Mandatory Course (Non-credit), **AEC**: Ability Enhancement Course, **SEC**: Skill Enhancement Course, **L:** Lecture, **T**: Tutorial, **P**: Practical **S= SDA**: Skill Development Activity, **CIE**: Continuous Internal Evaluation, **SEE**: Semester End Evaluation. K : This letter in the course code indicates common to all the stream of engineering. ESC: Engineering Science Course, ETC: Emerging Technology Course, PLC: Programming Language Course

.

Engineering Science Course (ESC/ETC/PLC)			
23CSI36A	Object oriented programming with java		
23CSI36B	Object oriented programming with C++		

Ability Enhancement Course – III (All are Laboratory Courses 0-0-2)			
23CSL38A	Data analytics with excel	23CSL38C	Project management with Git
23CSL38B	R Programming	23CSL38D	Data visualization with Python

Professional Core Course (IPCC): Refers to Professional Core Course Theory Integrated with practicals of the same course. Credit for IPCC can be 04 and its Teaching–Learning hours (L : T : P) can be considered as (3 : 0 : 2) or (2 : 2 : 2). The theory part of the IPCC shall be evaluated both by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper. For more details, the regulation governing the Degree of Bachelor of Engineering /Technology (B.E./B.Tech.) 2022-23 may please be referred.

National Service Scheme /Physical Education/Yoga: All students have to register for any one of the courses namely National Service Scheme (NSS), Physical Education (PE)(Sports and Athletics), and Yoga(YOG) with the concerned coordinator of the course during the first week of III semesters. Activities shall be carried out between III semester to the VI semester (for 4 semesters). Successful completion of the registered course and requisite CIE score is mandatory for the award of the degree. The events shall be appropriately scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities. These courses shall not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the course is mandatory for the award of degree.

NAGARJUNA COLLEGE OF ENGINEERING & TECHNOLOGY, BENGALURU

B.E. in Computer Science and Engineering

Scheme of Teaching and Examinations 2023

Outcome Based Education (OBE) and Choice Based Credit System (CBCS) (Effective from the academic year 2024-25)

IV SEMESTER

Sl. No	Course and Course Code		Course Title	Teaching Department (TD) and Question Paper Setting (PSB)	Teaching Hours /Week				Examination				Credits
					Theory Lecture	Tutorial	Practical	SDA	Duration in Exams	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1	PCC/BSC	23CST41	Analysis & design of algorithms	CSE	3	0	0		03	50	50	100	3
2	IPCC	23CSI42	Microcontrollers	CSE	3	0	2		03	50	50	100	4
3	IPCC	23CSI43	Database Management Systems	CSE	3	0	2		03	50	50	100	4
4	PCCL	23CSL44	Analysis & design of Algorithms lab	CSE	0	0	2		03	50	50	100	1
5	ESC	23CST45	ESC/ETC/PLC	CSE	2	2	0		03	50	50	100	3
6	AEC/ SEC	23CST/L46X	Ability Enhancement Course/Skill Enhancement Course- IV	CSE	If the Course is a Theory 1 If a Course is Laboratory 0	0 0	0 2		01 02	50	50	100	1
7	BSC	23BSC47	Biology For computer Engineers	TD / PSB: Any Branch/ BT, CHE,	2	0	0		03	50	50	100	2
8	UHV	23UHV48	Universal human values course	Any Department	1	0	0		01	50	50	100	2
9	MC		National Service Scheme (NSS)	NSS coordinator	0	0	2			100	---	100	0
			Physical Education (PE) (Sports and Athletics)	Physical Education Director									
			Yoga	Yoga Teacher									

Total						**500**	**400**	**900**	**19**

PCC: Professional Core Course, **PCCL**: Professional Core Course laboratory, **UHV**: Universal Human Value Course, **MC**: Mandatory Course (Non-credit), **AEC**: Ability Enhancement Course, **SEC**: Skill Enhancement Course, **L:** Lecture, **T**: Tutorial, **P**: Practical **S= SDA**: Skill Development Activity, **CIE**: Continuous Internal Evaluation, **SEE**: Semester End Evaluation. K : This letter in the course code indicates common to all the stream of engineering.

Ability Enhancement Course / Skill Enhancement Course - IV

23CST46A	Green IT and Sustainability	23CSL46C	UI/UX
23CST46B	Capacity planning for IT	23CSL46D	Technical writing using LATEX [Lab][0:0:2]

Engineering Science Course (ESC/ETC/PLC)

23CST45A	Discrete Mathematical structures	23CST45C	Optimization Technique
23CST45B	Graph Theory	23CST45D	Linear Algebra

Professional Core Course (IPCC): Refers to Professional Core Course Theory Integrated with practical of the same course. Credit for IPCC can be 04 and its Teaching–Learning hours (L : T : P) can be considered as (3 : 0 : 2) or (2 : 2 : 2). The theory part of the IPCC shall be evaluated both by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper. For more details, the regulation governing the Degree of Bachelor of Engineering /Technology (B.E./B.Tech.) 2022-23

National Service Scheme /Physical Education/Yoga: All students have to register for any one of the courses namely National Service Scheme (NSS), Physical Education (PE)(Sports and Athletics), and Yoga(YOG) with the concerned coordinator of the course during the first week of III semesters. Activities shall be carried out between III semester to the VI semester (for 4 semesters). Successful completion of the registered course and requisite CIE score is mandatory for the award of the degree. The events shall be appropriately scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities. These courses shall not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the courses is mandatory for the award of degree.

NAGARJUNA COLLEGE OF ENGINEERING & TECHNOLOGY, BENGALURU

B.E. in Computer Science and Engineering

Scheme of Teaching and Examinations 2022

Outcome Based Education (OBE) and Choice Based Credit System (CBCS) (Effective from the academic year 2024-25)

V SEMESTER

Sl. No	Course	Course Code	Course Title	Teaching Department (TD) and Question Paper Setting (PSB)	Teaching Hours /Week				Examination				Credits
					Theory Lecture	Tutorial	Practical	SDA	Duration in Exams	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1	PCC	22CST51	Operating System	CSE	3	0	0		03	50	50	100	3
2	IPCC	22CSI52	Computer Networks(IC)	CSE	3	0	2		03	50	50	100	4
3	PCC	22CST53	Theory of Computation	CSE	3	0	0		03	50	50	100	3
4	PEC	22CST54X	Professional Elective Course-I	CSE	3	0	0		03	50	50	100	3
5	PCCL	22CSL55	Web Technology Lab	CSE	0	0	2		03	50	50	100	1
6	PCCL	22CSL56	Python Library Lab	CSE	0	0	2		03	50	50	100	1
7	PROJ	22CSP57	Mini Project	CSE	0	0	4		03	50	50	100	2

8	AEC/ SEC	22CST58	Research Methodology and IPR	Any Department	2	2	0		02/03	50	50	100	3
9	MC	22CST59	Environmental Studies	CSE	2	0	0		02	50	50	100	2
10	MC	22CSX59X	National Service Scheme (NSS)	NSS Coordinator	0	0	2			100	---	100	0
		22CSX59X	Physical Education (PE) (Sports and Athletics)	Physical Education Director									
		22CSX59X	Yoga	Yoga Teacher									
									Total	**550**	**350**	**900**	**22**

PCC: Professional Core Course, **PCCL**: Professional Core Course laboratory, **UHV**: Universal Human Value Course, **MC**: Mandatory Course (Non-credit), **AEC**: Ability Enhancement Course, **SEC**: Skill Enhancement Course, **L:** Lecture, **T**: Tutorial, **P**: Practical **S= SDA**: Skill Development Activity, **CIE**: Continuous Internal Evaluation, **SEE**: Semester End Evaluation. K : This letter in the course code indicates common to all the stream of engineering. ESC: Engineering Science Course, ETC: Emerging Technology Course, PLC: Programming Language Course

.

Professional Elective Course (PEC)

22CST541	**Distributed Systems**	22CST543	Unix System Programming
22CST542	Artificial Intelligence	22CST544	**Data Analytics/Computer Graphics/Data Science**

Professional Core Course (IPCC): Refers to Professional Core Course Theory Integrated with practical's of the same course. Credit for IPCC can be 04 and its Teaching–Learning hours (L : T : P) can be considered as (3 : 0 : 2) or (2 : 2 : 2). The theory part of the IPCC shall be evaluated both by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper. For more details, the regulation governing the Degree of Bachelor of Engineering /Technology (B.E./B.Tech.) 2022-23 may please be referred.

National Service Scheme /Physical Education/Yoga: All students have to register for any one of the courses namely National Service Scheme (NSS), Physical Education (PE)(Sports and Athletics), and Yoga(YOG) with the concerned coordinator of the course during the first week of III semesters. Activities shall be carried out between III semester to the VI semester (for 4 semesters). Successful completion of the registered course and requisite CIE score is mandatory for the award of the degree. The events shall be appropriately scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities. These courses shall not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the course is mandatory for the award of degree.

NAGARJUNA COLLEGE OF ENGINEERING & TECHNOLOGY, BENGALURU

B.E. in Computer Science and Engineering

Scheme of Teaching and Examinations 2022

Outcome Based Education (OBE) and Choice Based Credit System (CBCS) (Effective from the academic year 2024-25)

VI SEMESTER

Sl. No	Course	Course Code	Course Title	Teaching Department (TD) and Question Paper Setting (PSB)	Teaching Hours /Week				Duration in Exams	Examination			Credits
					Theory Lecture	Tutorial	Practical	SDA		CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1	IPCC	22CSI61	Cloud Computing	CSE	3	0	2		03	50	50	100	4
2	PCC	22CST62	Machine Learning	CSE	4	0	0		03	50	50	100	4
3	PEC	22CST63X	Professional Elective Course-II	CSE	3	0	0		03	50	50	100	3
4	OEC	22CST64	Open Elective Course-I	CSE	3	0	0		03	50	50	100	3
5	PROJ	22CSP65	Project Phase I	CSE	0	0	4		03	100	--	100	2
6	PCCL	22CSL66	Machine Learning Lab	CSE	0	0	2		03	50	50	100	1
7	AEC/SDC	22CS67X/ 22CS67LX	Ability Enhancement Course/Skill Development Course V	Concerned Board	If offered as Theory courses				01	50	50	100	1
					1	0	0						
					If offered as Practical courses								
					0	0	2						
8	MC		National Service Scheme (NSS)	NSS coordinator	0	0	2			100	--	100	0
			Physical Education (PE) (Sports and	Physical									

			Athletics)	Education Director									
			Yoga	Yoga Teacher									
									Total	**500**	**300**	**800**	**18**

PCC: Professional Core Course, **PCCL**: Professional Core Course laboratory, **UHV**: Universal Human Value Course, **MC**: Mandatory Course (Non-credit), **AEC**: Ability Enhancement Course, **SEC**: Skill Enhancement Course, **L:** Lecture, **T**: Tutorial, **P**: Practical **S= SDA**: Skill Development Activity, **CIE**: Continuous Internal Evaluation, **SEE**: Semester End Evaluation. K: This letter in the course code indicates that it is common to all engineering streams. ESC: Engineering Science Course, ETC: Emerging Technology Course, PLC: Programming Language Course.

Professional Elective Course (PEC)			
22CST631	Blockchain Technology	22CST633	Compiler Design
22CST632	**Digital Image Processing(DIP)**	22CST634	**Soft Computing**

Ability Enhancement Course/ Skill Development Course – V			
22CS671	Progressive App Development	22CS673	DevOps
22CS672	**Tosca- Automated Software Testing**	22CS674	**Agile**

Professional Core Course (IPCC): Refers to Professional Core Course Theory Integrated with practical of the same course. Credit for IPCC can be 04 and its Teaching–Learning hours (L : T : P) can be considered as (3 : 0 : 2) or (2 : 2 : 2). The theory part of the IPCC shall be evaluated both by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper. For more details, the regulation governing the Degree of Bachelor of Engineering /Technology (B.E./B.Tech.) 2022-23 may please be referred.

National Service Scheme /Physical Education/Yoga: All students have to register for any one of the courses, namely National Service Scheme (NSS), Physical Education (PE)(Sports and Athletics), and Yoga(YOG), with the concerned coordinator of the course during the first week of III semesters. Activities shall be carried out between III and VI semesters (for 4 semesters). Successful completion of the registered course and requisite CIE score is mandatory for the award of the degree. The colleges shall appropriately schedule the events, and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities. These courses shall not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the course is mandatory for the award of a degree.

Open Elective Courses

22CST641	Introduction to Data Structures	22CST643	Introduction to AI
22CST642	Fundamentals of Operating Systems	22CST644	Introduction to Web Programming

NAGARJUNA COLLEGE OF ENGINEERING & TECHNOLOGY, BENGALURU

B.E. in Computer Science and Engineering

Scheme of Teaching and Examinations 2022

Outcome Based Education (OBE) and Choice Based Credit System (CBCS) (Effective from the academic year 2024-25)

VII SEMESTER

Sl. No	Course	Course Code	Course Title	Teaching Department (TD) and Question Paper Setting (PSB)	Teaching Hours /Week				Duration in Exams	Examination			Credits
					Theory Lecture	Tutorial	Practical	SDA		CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1	IPCC	22CSI71	Internet of Things(IC)	CSE	3	0	2		03	50	50	100	4
2	IPCC	22CSI72	Natural Language Processing (NLP) (IC)	CSE	3	0	2		03	50	50	100	4
3	PCC	22CST73	Cryptography & Network Security	CSE	4	0	0		03	50	50	100	4
4	PEC	22CST74X	Professional Elective Course- III	CSE	3	0	0		03	50	50	100	3
5	OEC	22CST75X	Open Elective Course-II	CSE	3	0	0		03	50	50	100	3

6	PROJ	22CSP76	Major Project Phase-2	CSE	0	0	12		03	100	100	200	6
									Total	350	350	700	**24**

Professional Elective Course (PEC)			
22CST741	**Predictive Analytics**	22CST743	**Big Data Analytics**
22CST742	**Data Visualization with Tableau**	22CST744	**Deep Learning(DL)**
Open Elective Course (OEC)			
22CST751	Introduction to DBMS	22CST753	Software Engineering
22CST752	Computing Paradigms	22CST754	Cloud Computing

PCC: Professional Core Course, **PCCL**: Professional Core Course laboratory, **PEC**: Professional Elective Course, **OEC**: Open Elective Course PR: Project Work, **L:** Lecture, **T**: Tutorial, **P**: Practical **S= SDA**: Skill Development Activity, **CIE**: Continuous Internal Evaluation, **SEE**: Semester End Evaluation. **TD-** Teaching Department, **PSB**: Paper Setting department, **OEC**: Open Elective Course, **PEC**: Professional Elective Course. **PROJ**: Project work

Note: VII and VIII semesters of IV years of the program

(1) Institutions can swap the VII and VIII Semester Schemes of Teaching and Examinations to accommodate research internships/ industry internships after the VI semester**.**

(2) Credits earned for the courses of VII and VIII Semester Scheme of Teaching and Examinations shall be counted against the corresponding semesters whether the VII or VIII semesters is completed during the beginning of the IV year or the later part of IV years of the program.

Professional Elective Courses (PEC): A professional elective (PEC) course is intended to enhance the depth and breadth of educational experience in the Engineering and Technology curriculum. Multidisciplinary courses that are added supplement the latest trend and advanced technology in the selected stream of engineering. Each group will provide an option to select one course. The minimum number of students' strengths for offering professional electives is 10. However, this conditional

shall not be applicable to cases where the admission to the program is less than 10.

Open Elective Courses:

Students belonging to a particular stream of Engineering and Technology are not entitled to the open electives offered by their parent Department. However, they can opt for an elective offered by other Departments, provided they satisfy the prerequisite condition if any. Registration to open electives shall be documented under the guidance of the Program Coordinator/ Advisor/Mentor. The minimum numbers of students' strength for offering Open Elective Course is 10. However, this condition shall not be applicable to class where the admission to the program is less than 10.

PROJECT WORK (21CSP75): The objective of the Project work is

(i) To encourage independent learning and the innovative attitude of the students.

(ii) To develop interactive attitude, communication skills, organization, time management, and presentation skills.

(iii) To impart flexibility and adaptability.

(iv) To inspire team working.

(v) To expand intellectual capacity, credibility, judgment and intuition.

(vi) To adhere to punctuality, setting and meeting deadlines.

(vii) To install responsibilities to oneself and others.

(viii) To train students to present the topic of project work in a seminar without any fear, face the audience confidently, enhance communication skills, involve in group discussion to present and exchange ideas.

CIE procedure for Project Work:

(1) Single discipline: The CIE marks shall be awarded by a committee consisting of the Head of the concerned Department and two senior faculty members of the Department, one of whom shall be the Guide.

The CIE marks awarded for the project work, shall be based on the evaluation of the project work Report, project presentation skill, and question and answer session in the ratio 50:25:25. The marks awarded for the project report shall be the same for all the batch mates. **(2) Interdisciplinary:** Continuous Internal Evaluation shall be group-wise at the college level with the participation of all guides of the college. Participation of external guide/s, if any, is desirable. The CIE marks awarded for the project work, shall be based on the evaluation of project work Report, project presentation skill, and question and answer session in the ratio 50:25:25. The marks awarded for the project report shall be the same for all the batch mates. **SEE procedure for Project Work:** SEE for project work will be conducted by the two examiners appointed by the University. The SEE marks awarded for the project work shall be based on the evaluation of project work Report, project presentation skill, and question and answer session in the ratio 50:25:25.

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

B.E. in Computer Science and Engineering

Scheme of Teaching and Examinations2022

Outcome Based Education (OBE) and Choice Based Credit System (CBCS)

(Effective from the academic **year 2024-25**)

VIII SEMESTER (Swappable VII and VIII SEMESTER)

Sl. No	Course and Course Code		Course Title	Teaching Department (TD) and Question Paper Setting Board (PSB)	Teaching Hours /Week				Examination				Credits
					Theory Lecture	Tutorial	Practical	SDA	Duration in hrs	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1	PEC	22CSP81X	Professional Elective (Online course) only through NPTEL	CSE	3	0	0		03	50	50	100	3
2	OEC	22CSP82X	Open Elective (Online course) only through NPTEL	CSE	3	0	0		01	50	50	100	3
3	INT	22CSP83	**Internship (Industry/Research) (14 - 20 weeks)**		0	0	12		03	100	100	200	10
										200	**200**	**400**	**16**
Professional Elective Course (Online courses)													
Open Elective Courses (Online Courses)													

L: Lecture, **T**: Tutorial, **P**: Practical **S= SDA**: Skill Development Activity, **CIE**: Continuous Internal Evaluation, **SEE**: Semester End Evaluation. **TD-** Teaching Department, **PSB**: Paper Setting department, **OEC**: Open Elective Course, **PEC**: Professional Elective Course. **PROJ**: Project work, **INT**: Industry Internship / Research Internship / Rural Internship

Note: VII and VIII semesters of IV years of the program Swapping Facility

- Institutions can swap VII and VIII Semester Scheme of Teaching and Examinations to accommodate **research internships/ industry internships/Rural Internship**

after the VI semester**.**

- Credits earned for the courses of VII and VIII Semester Scheme of Teaching and Examinations shall be counted against the corresponding semesters whether VII or VIII semester is completed during the beginning of IV year or later part of IV year of the program.
- **Note: For BCS801x and BCS802x courses BOS will announce the list of courses in 6th , 7th & 8th Sem . Students can register in any of the semesters to earn the credits in 8th Sem.**

Elucidation:

At the beginning of IV years of the program i.e., after VI semester, VII semester classwork and VIII semester **Research Internship /Industrial Internship / Rural Internship** shall be permitted to be operated simultaneously by the University so that students have ample opportunity for an internship. In other words, a good percentage of the class shall attend VII semester classwork and a similar percentage of others shall attend to Research Internship or Industrial Internship or Rural Internship.

Research/Industrial /Rural Internship shall be carried out at an Industry, NGO, MSME, Innovation center, Incubation center, Start-up, center of Excellence (CoE), Study Centre established in the parent institute and /or at reputed research organizations/institutes.

The mandatory Research internship /Industry internship / Rural Internship is for 14 to 20 weeks. The internship shall be considered as a head of passing and shall be considered for the award of a degree. Those, who do not take up/complete the internship shall be declared to fail and shall have to complete it during the subsequent University examination after satisfying the internship requirements.

Research internship: A research internship is intended to offer the flavor of current research going on in the research field. It helps students get familiarized with the field and imparts the skill required for carrying out research.

Industry internship: Is an extended period of work experience undertaken by students to supplement their degree for professional development. It also helps them learn to overcome unexpected obstacles and successfully navigate organizations, perspectives, and cultures. Dealing with contingencies helps students recognize, appreciate, and adapt to organizational realities by tempering their knowledge with practical constraints.

Rural Internship: Rural development internship is an initiative of Unnat Bharat Abhiyan Cell, RGIT in association with AICTE to involve students of all departments studying in different academic years for exploring various opportunities in techno-social fields, to connect and work with Rural India for their upliftment.

The faculty coordinator or mentor has to monitor the student's internship progress and interact with them to guide for the successful completion of the internship. The students are permitted to carry out the internship anywhere in India or abroad. University shall not bear any expenses incurred in respect of the internship.

With the consent of the internal guide and Principal of the Institution, students shall be allowed to carry out the internship at their hometown (**within or outside the state or abroad**), provided favorable facilities are available for the internship and the student remains regularly in contact with the internal

guide. **University shall not bear any cost involved in carrying out the internship by students.** However, students can receive any financial assistance extended by the organization.

Professional Elective /Open Elective Course: These are ONLINE courses suggested by the respective Board of Studies. Details of these courses shall be made available for students on the VTU web portal.

Please note: If any clarifications / suggestions please email to**sbhvtuso@yahoo.com**

III SEMESTER

COURSE SYLLABUS

OPERATING SYSTEMS					
Course Code	**L:T:P:S**	**Credits**	**Exam Marks**	**Exam Duration**	**Course Type**
22CSI33	3:0:2:0	4	CIE:50 SEE:50	3 Hours	PCC/BSC

PREREQUISITES:

Good knowledge of C, Computer Organization and Architecture, x86 Assembly level programming.

Course Objectives :

The Student will be able to:

- To Demonstrate the need for OS and different types of OS
- To discuss suitable techniques for management of different resources
- To demonstrate different APIs/Commands related to processor, memory, storage and file system management.

Teaching-Learning Process (General Instructions)

Teachers can use the following strategies to accelerate the attainment of the various course outcomes.

1. Lecturer methods (L) need not to be only traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
5. Role-play for process scheduling.
6. Demonstrate the installation of any one Linux OS on VMware/Virtual Box

Syllabus

Module I

Introduction to operating systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments.

Operating System Services: User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System debugging, Operating System generation; System boot.

Textbook 1: Chapter – 1 (1.1-1.12), 2 (2.2-2.11) **8 Hours**

Module II

Process Management: Process concept; Process scheduling; Operations on processes; Inter process communication.

Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues.
Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Thread scheduling; Multiple-processor scheduling,

Textbook 1: Chapter – 3 (3.1-3.4), 4 (4.1-4.4), 5 (5.1 -5.5) **8 Hours**

Module III

Process Synchronization: Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization.

Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

Textbook 1: Chapter – 6 (6.1-6.6), 7 (7.1 -7.7) **8 Hours**

Module IV

Memory Management: Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.

Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing.

Textbook 1: Chapter -8 (8.1-8.6), 9 (9.1-9.6) **8 Hours**

Module V

File System, Implementation of File System: File system: File concept; Access methods; Directory and Disk structure; File system mounting; File sharing; **Implementing File system:** File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.

Secondary Storage Structure, Protection: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; **Protection**: Goals of protection, Principles of protection, Domain of protection, Access matrix.

Textbook 1: Chapter – 10 (10.1-10.5) ,11 (11.1-11.5),12 (12.1-12.5), 14 (14.1-14.4) **8 Hours**

PRACTICAL COMPONENT OF IPCC

Sl. NO.	Experiments
1	Develop a c program to implement the Process system calls (fork (), exec(), wait(), create process, terminate process)
2	Simulate the following CPU scheduling algorithms to find turnaround time and waiting time a) FCFS b) SJF c) Round Robin d) Priority.
3	Develop a C program to simulate producer-consumer problem using semaphores.
4	Develop a C program which demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in your program.
5	Develop a C program to simulate Bankers Algorithm for DeadLock Avoidance.
6	Develop a C program to simulate the following contiguous memory allocation Techniques: a) Worst fit b) Best fit c) First fit.
7	Develop a C program to simulate page replacement algorithms:

	a) FIFO b) LRU
8	Simulate following File Organization Techniques a) Single level directory b) Two level directory
9	Develop a C program to simulate the Linked file allocation strategies.
10	Develop a C program to simulate SCAN disk scheduling algorithm.

Course outcomes:

The Student will be able to:

CO 1. Explain the structure and functionalities of operating system.

CO 2. Apply appropriate algorithms for the given problem.

CO 3. Analyze the various techniques used in operating system design and implementation.

CO 4. Compare various types of operating system based on different attributes.

CO 5. Prepare a comparative report on functionalities of operating system.

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 8th edition, Wiley-India, 2015

Reference Books:

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

E-Resources:

1. https://youtu.be/mXw9ruZaxzQ
2. https://youtu.be/vBURTt97EkA
3. https://www.tutorialspoint.com/operating_system/index.htm.
4. https://www.studytonight.com/operating-system/.

CO-PO-PSO Mapping

PO'S	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO1 0	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO2	3	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO3	-	3	-	-	-	-	-	-	-	-	-	-	3	-	-
CO4	-	-	2	-	-	-	-	-	-	-	-	-	3	-	-
CO5	-	-	-	-	-	-	-	-	-	2	-	2	3	-	-
CO	3	3	2							2		2	3	-	-

	Object Oriented Programming with JAVA		Semester	3
	Course Code	23CST36A	CIE Marks	50
	Teaching Hours/Week (L: T:P: S)	2:0:2	SEE Marks	50
	Total Hours of Pedagogy	28 Hours of Theory + 20 Hours of Practical	Total Marks	100
	Credits	03	Exam Hours	03
	Examination type (SEE)	Theory		

Course objectives:

- To learn primitive constructs JAVA programming language.
- To understand Object Oriented Programming Features of JAVA.
- To gain knowledge on: packages, multithreaded programing and exceptions.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes and make Teaching –Learning more effective

1. Use Online Java Compiler IDE: https://www.jdoodle.com/online-java-compiler/ or any other.
2. Demonstration of programing examples.
3. Chalk and board, power point presentations
4. Online material (Tutorials) and video lectures.

Module-1

An Overview of Java: Object-Oriented Programming (Two Paradigms, Abstraction, The Three OOP Principles), Using Blocks of Code, Lexical Issues (Whitespace, Identifiers, Literals, Comments, Separators, The Java Keywords).

Data Types, Variables, and Arrays: The Primitive Types (Integers, Floating-Point Types, Characters, Booleans), Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays, Introducing Type Inference with Local Variables.

Operators: Arithmetic Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses.

Control Statements: Java's Selection Statements (if, The Traditional switch), Iteration Statements (while, do-while, for, The For-Each Version of the for Loop, Local Variable Type Inference in a for Loop, Nested Loops), Jump Statements (Using break, Using continue, return).

Chapter 2, 3, 4, 5

Module-2

Introducing Classes: Class Fundamentals**,** Declaring Objects**,** Assigning Object Reference Variables**,** Introducing Methods**,** Constructors**,** The this Keyword**,** Garbage Collection.

Methods and Classes: Overloading Methods, Objects as Parameters, Argument Passing, Returning Objects, Recursion, Access Control, Understanding static, Introducing final, Introducing Nested and Inner Classes.

Chapter 6, 7

Module-3

	Inheritance: Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, Local Variable Type Inference and Inheritance, The Object Class. **Interfaces:** Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods. **Chapter 8, 9**

	Module-4
	Packages: Packages, Packages and Member Access, Importing Packages. **Exceptions:** Exception-Handling Fundamentals**,** Exception Types**,** Uncaught Exceptions**,** Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions**.** **Chapter 9, 10**
	Module-5
	Multithreaded Programming: The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities, Synchronization, Interthread Communication, Suspending, Resuming, and Stopping Threads, Obtaining a Thread's State. **Enumerations, Type Wrappers and Autoboxing:** Enumerations (Enumeration Fundamentals**,** The values() and valueOf() Methods), Type Wrappers (Character, Boolean, The Numeric Type Wrappers), Autoboxing (Autoboxing and Methods, Autoboxing/Unboxing Occurs in Expressions, Autoboxing/Unboxing Boolean and Character Values). **Chapter 11, 12**

Course outcome (Course Skill Set)

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

1. Explain the basics concepts of Java.
2. Apply the concepts of java in solving a given problems.
3. Analyze the given scenario to provide feasible solution.
4. Design and develop solution for any real world /complex problem using java packages.
5. Demonstrate the given experiments by using tools of java.

Programming Experiments (Suggested and are not limited to)

1. Develop a JAVA program to add TWO matrices of suitable order N (The value of N should be read from command line arguments).
2. Develop a stack class to hold a maximum of 10 integers with suitable methods. Develop a JAVA main method to illustrate Stack operations.
3. A class called Employee, which models an employee with an ID, name and salary, is designed as shown in the following class diagram. The method raiseSalary (percent) increases the salary by the given percentage. Develop the Employee class and suitable main method for demonstration.
4. A class called MyPoint, which models a 2D point with x and y coordinates, is designed as follows:
   - Two instance variables x (int) and y (int).
   - A default (or "no-arg") constructor that construct a point at the default location of (0, 0).
   - A overloaded constructor that constructs a point with the given x and y coordinates.
   - A method setXY() to set both x and y.
   - A method getXY() which returns the x and y in a 2-element int array.
   - A toString() method that returns a string description of the instance in the format "(x, y)".
   - A method called distance(int x, int y) that returns the distance from this point to another point at the given (x, y) coordinates
   - An overloaded distance(MyPoint another) that returns the distance from this point to the given MyPoint instance (called another)
   - Another overloaded distance() method that returns the distance from this point to the origin (0,0) Develop the code for the class MyPoint. Also develop a JAVA program (called TestMyPoint) to test all the methods defined in the class.
5. Develop a JAVA program to create a class named shape. Create three sub classes namely: circle, triangle and square, each class has two member functions named draw () and erase (). Demonstrate

polymorphism concepts by developing suitable methods, defining member data and main program.

6. Develop a JAVA program to create an abstract class Shape with abstract methods calculateArea() and calculatePerimeter(). Create subclasses Circle and Triangle that extend the Shape class and implement the respective methods to calculate the area and perimeter of each shape.
7. Develop a JAVA program to create an interface Resizable with methods resizeWidth(int width) and resizeHeight(int height) that allow an object to be resized. Create a class Rectangle that implements the Resizable interface and implements the resize methods
8. Develop a JAVA program to create an outer class with a function display. Create another class inside the outer class named inner with a function called display and call the two functions in the main class.
9. Develop a JAVA program to raise a custom exception (user defined exception) for DivisionByZero using try, catch, throw and finally.
10. Develop a JAVA program to create a package named mypack and import & implement it in a suitable class.
11. Write a program to illustrate creation of threads using runnable class. (start method start each of the newly created thread. Inside the run method there is sleep() for suspend the thread for 500 milliseconds).
12. Develop a program to create a class MyThread in this class a constructor, call the base class constructor, using super and start the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

CIE for the theory component of the IPCC (maximum marks 50)

- IPCC means practical portion integrated with the theory of the course.
- CIE marks for the theory component are **25 marks** and that for the practical component is **25 marks**.
- 25 marks for the theory component are split into **15 marks** for two Internal Assessment Tests (Two Tests, each of 15 Marks with 01-hour duration, are to be conducted) and **10 marks** for other assessment methods mentioned in 22OB4.2. The first test at the end of 40-50% coverage of the syllabus and the second test after covering 85-90% of the syllabus.
- Scaled-down marks of the sum of two tests and other assessment methods will be CIE marks for the theory component of IPCC (that is for **25 marks)**.
- The student has to secure 40% of 25 marks to qualify in the CIE of the theory component of IPCC.

CIE for the practical component of the IPCC

- **15 marks** for the conduction of the experiment and preparation of laboratory record, and **10 marks** for the test to be conducted after the completion of all the laboratory sessions.
- On completion of every experiment/program in the laboratory, the students shall be evaluated including viva-voce and marks shall be awarded on the same day.
- The CIE marks awarded in the case of the Practical component shall be based on the continuous evaluation of the laboratory report. Each experiment report can be evaluated for 10 marks. Marks of all experiments' write-ups are added and scaled down to **15 marks**.
- The laboratory test **(duration 02/03 hours)** after completion of all the experiments shall be conducted for 50 marks and scaled down to **10 marks.**
- Scaled-down marks of write-up evaluations and tests added will be CIE marks for the laboratory component of IPCC for **25 marks**.
- The student has to secure 40% of 25 marks to qualify in the CIE of the practical component of the IPCC.

SEE for IPCC

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the course (**duration 03 hours**)

1. The question paper will have ten questions. Each question is set for 20 marks.
2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.
3. The students have to answer 5 full questions, selecting one full question from each module.
4. Marks scored by the student shall be proportionally scaled down to 50 Marks

The theory portion of the IPCC shall be for both CIE and SEE, whereas the practical portion will have aCIE component only. Questions mentioned in the SEE paper may include questions from the practical component.

Suggested Learning Resources:

Textbook

1. Java: The Complete Reference, Twelfth Edition, by Herbert Schildt, November 2021, McGraw-Hill, ISBN: 9781260463422

Reference Books

1. Programming with Java, 6th Edition, by E Balagurusamy, Mar-2019, McGraw Hill Education, ISBN:9789353162337.
2. Thinking in Java, Fourth Edition, by Bruce Eckel, Prentice Hall, 2006 (https://sd.blackball.lv/library/thinking_in_java_4th_edition.pdf)

Web links and Video Lectures (e-Resources):

- Java Tutorial: https://www.geeksforgeeks.org/java/
- Introduction To Programming In Java (by Evan Jones, Adam Marcus and Eugene Wu): https://ocw.mit.edu/courses/6-092-introduction-to-programming-in-java-january-iap-2010/
- Java Tutorial: https://www.w3schools.com/java/
- Java Tutorial: https://www.javatpoint.com/java-tutorial

Activity Based Learning (Suggested Activities)/ Practical Based learning

1. Installation of Java (Refer: https://www.java.com/en/download/help/index_installing.html)
2. Demonstration of online IDEs like geeksforgeeks, jdoodle or any other Tools
3. Demonstration of class diagrams for the class abstraction, type visibility, composition andinheritance

Assessment Method

- Programming Assignment / Course Project

Project Management with Git

Course Code	**23CSL38C**	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	0: 0 : 2: 0	SEE Marks	50
Credits	01	Exam Marks	100
Examination type (SEE)	Practical	Exam Hours	2 hrs

Course objectives:

- .To familiar with basic command of Git
- To create and manage branches
- To understand how to collaborate and work with Remote Repositories
- To familiar with virion controlling commands

Sl.NO	Experiments
1	**Setting Up and Basic Commands** Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.
2	**Creating and Managing Branches** Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."
3	**Creating and Managing Branches** Write the commands to stash your changes, switch branches, and then apply the stashed changes.
4	**Collaboration and Remote Repositories** Clone a remote Git repository to your local machine.
5	**Collaboration and Remote Repositories** Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.
6	**Collaboration and Remote Repositories** Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge.
7	**Git Tags and Releases** Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.
8	**Advanced Git Operations**

	Write the command to cherry-pick a range of commits from "source-branch" to the current branch.
9	**Analysing and Changing Git History** Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?
10	**Analysing and Changing Git History** Write the command to list all commits made by the author "JohnDoe" between "2023-01-01" and "2023-12-31."
11	**Analysing and Changing Git History** Write the command to display the last five commits in the repository's history.
12	**Analysing and Changing Git History** Write the command to undo the changes introduced by the commit with the ID "abc123".

Course outcomes (Course Skill Set):

At the end of the course the student will be able to:

- Use the basics commands related to git repository
- Create and manage the branches
- Apply commands related to Collaboration and Remote Repositories
- Use the commands related to Git Tags, Releases and advanced git operations
- Analyse and change the git history

IV SEMESTER

COURSE SYLLABUS

Analysis and Design of Algorithms

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
23CST41	3:0:0:0	3	CIE: 50 SEE: 50	3 Hours	PCC

Description of the course: Algorithms are the soul of computing. Algorithmic thinking is rooted in ancient mathematics. This course introduces basic methods for the design and analysis of efficient algorithms emphasizing methods useful in practice. Different algorithms for a given computational task are presented and their relative merits evaluated based on performance measures. The following important computational problems will be discussed: Sorting, Searching, String matching elements of Dynamic programming and Greedy algorithms, Graph algorithms (Shortest path, Spanning trees, Tree traversals). This course covers principles of algorithm design, elementary analysis of algorithms, and fundamental data structures. The emphasis is on choosing appropriate data structures and designing correct and efficient algorithms to operate on these data structures.

Prerequisite: Data Structures, Discrete Mathematics, Graph Theory.

Course Objectives:
This course will enable a student to:

- Acquire the knowledge of Algorithm and problem solving technique.
- Learn how to analyze the complexity of an algorithm in terms of time and space.
- Understand techniques like divide and conquer, decrease and conquer, transfer and conquer to solve problems.
- Understand Space-Time Trade-offs and Dynamic programming technique.
- Describe the limitations of algorithms.

Syllabus

Module – I

Introduction: Definition of an algorithm, Fundamentals of algorithmic problem solving.

Fundamentals of the analysis of algorithm efficiency: Asymptotic Notations and basic efficiency classes, Mathematical Analysis of Non-Recursive and Recursive Algorithms.

Brute Force Approaches: Selection Sort, Bubble Sort, Sequential Search and Brute Force String Matching. **08 Hours**

Module – II

Divide and Conquer: Merge Sort, Quick Sort.

The Greedy Method: Prim's Algorithm, Kruskal's Algorithm, Single Source Shortest Paths, Huffman Trees and Codes.

Exhaustive Search: The Travelling Salesperson problem, Knapsack Problem. **08 Hours**

Module – III

Exhaustive Search (Contd.): Depth First Search, Breadth First Search.

Decrease and Conquer: Insertion Sort, Topological Sorting, Binary Search.
Transfer and Conquer: Balanced Search Trees, Heap (Top-down and Bottom-Up Heap construction) and Heap sort. **08 Hours**

Module – IV

Space-Time Trade-offs: Sorting by Counting, Input Enhancement in String Matching (Horspool's algorithm), Open Hashing, Closed Hashing.

Dynamic programming: Knapsack Problem, Warshall's Algorithm, Floyd's Algorithm for the All-Pairs Shortest Paths Problem. **08 Hours**

Module – V

Limitations of Algorithm Power: Decision Trees, P, NP, and NP-Complete Problems.

Backtracking: n-Queens problem, Subset–Sum Problem.

Branch and Bound: Assignment problem, Knapsack problem. **08 Hours**

Course Outcomes:
After studying this course, the students will be able to:

- Explain the concepts of algorithms and problem solving techniques.
- Apply various algorithmic techniques or methods to solve Computer Science and Engineering related problems.
- Perform amortize analysis for algorithm and differentiate among algorithms with respect to time efficiency and space efficiency.
- Design and develop solutions for the problems using algorithms.

Text Book:

1. Anany Levitin: "Introduction to The Design and Analysis of Algorithms", (Chapter 1.1, 1.2, 2.1, 2.2, 2.3, 2.4, 2.5, 3.1, 3.2, 3.4, 3.5, 4.1, 4.2, 4.4, 5.1, 5.2, 6.3, 6.4, 7.1, 7.2, 7.3, 8.2, 8.4, 9.1, 9.2, 9.3, 9.4, 11.2, 11.3, 12.1, 12.2), Pearson Education, Delhi, 3rd Edition, 2007, ISBN-13 : 9780132316811.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein: "Introduction to Algorithms", PHI, London, England, 3rd Edition, 2010, ISBN-13: 9780262033848.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: "Fundamentals of Computer Algorithms", (Chapters 1,3-8,10-12), Universities Press, Hyderabad, 2nd Edition, 2007, ISBN-10: 8173716129.
3. R.C.T. Lee, S.S. Tseng, R.C. Chang and Y.T. Tsai: "Introduction to the Design and Analysis of Algorithms A Strategic Approach", McGraw-Hill Higher Education, USA, 3rd International Edition, 2005, ISBN-13: 978-0071243469.

Reference Online Resources:

1. https://www.geeksforgeeks.org/fundamentals-of-algorithms/

2. http://www.citc.ui.ac.ir/zemoni/cls.pdf

COURSE EVALUATION SCHEME:

Theory Component	IAT-1 after completion 45 to 50% Syllabus	25 Marks
	IAT-2 after completion 95 to 100% Syllabus	25 Marks
	Sum of two IATs	**50 Marks**
	Total 25 Marks : Reduced to 25 Marks	
	CCE-1	25 Marks
	CCE-2	25 Marks
	Sum of two CCEs	**50 Marks**
	Total 25 Marks : Reduced to 25 Marks	
Grand Total of CIE Marks		**50 Marks**
SEE Marks		**50 Marks**
Total Marks (CIE + SEE)		**100 Marks**

CO-PO MAPPING:

POS / COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	-	-	-	-	-	-	-	-	-	-	1	-	1	-
CO2	-	3	-	-	-	-	-	-	-	-	-	2	2	2	2
CO3	-	3	-	-	-	-	-	-	-	-	-	2	2	2	2
CO4	-	-	3	2	-	-	-	-	-	-	-	2	2	2	2

Analysis and Design of Algorithms

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
23CSL44	0:0:2:0	1	CIE: 50 SEE: 50	3 Hours	PCC

Prerequisite: Data Structures, Discrete Mathematics, Graph Theory.

Course Objectives:

This course will enable a student to:

- Acquire the knowledge of Algorithm and problem solving technique.
- Learn how to analyze the complexity of an algorithm in terms of time and space.
- Understand techniques like divide and conquer, decrease and conquer, transfer and conquer to solve problems.
- Understand Space-Time Trade-offs and Dynamic programming technique.
- Describe the limitations of algorithms.

List of Lab Programs:

Design, develop and implement the specified algorithms for the following problems using C/C++ Language in LINUX environment.

1. Consider a list of 'n' files numbered using ID's. Write a C program to sort files based on its ID using Quick sort method.
2. Consider a list of 'n' books numbered using Book_ID's. Write a C program to sort files based on its Book_ID using Merge sort method.
3. Consider a Electrical layout where 'n' houses are connected by electrical wires. Design a 'C' program using Prim's algorithm to output a connection with minimum cost.
4. Suppose a travel agent is interested in finding shortest path from a single city to all the other cities in a network of 'n' cities. Write a C program to implement this using Djikstra's algorithm.
5. Consider a network having 'n' systems. Design a DFS based program in 'C' which outputs all systems reachable from a given system.
6. Suppose in a network of cities, you are interested in finding shortest paths between all cities. Design a 'C' program to implement this using Floyd's algorithm.
7. Implement 0/1 Knapsack problem using Dynamic Programming.
8. Consider 'N' patients and 'N X N' small rooms. Design a C program to allot the patients to these rooms using N-queen's method such that no two patients are allotted rooms in same row, column or diagonal.

Course Outcomes:
After studying this course, the students will be able to:

- Explain the concepts of algorithms and problem solving techniques.
- Apply various algorithmic techniques or methods to solve Computer Science and Engineering related problems.
- Perform amortize analysis for algorithm and differentiate among algorithms with respect to time efficiency and space efficiency.

- Design and develop solutions for the problems using algorithms.

Text Book:

2. Anany Levitin: "Introduction to The Design and Analysis of Algorithms", Pearson Education, Delhi, 3rd Edition, 2007, ISBN-13 : 9780132316811.

Reference Books:

4. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein: "Introduction to Algorithms", PHI, London, England, 3rd Edition, 2010, ISBN-13: 9780262033848.
5. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: "Fundamentals of Computer Algorithms", (Chapters 1,3-8,10-12), Universities Press, Hyderabad, 2nd Edition, 2007, ISBN-10: 8173716129.
6. R.C.T. Lee, S.S. Tseng, R.C. Chang and Y.T. Tsai: "Introduction to the Design and Analysis of Algorithms A Strategic Approach", McGraw-Hill Higher Education, USA, 3rd International Edition, 2005, ISBN-13: 978-0071243469.

Reference Online Resources:

3. https://www.geeksforgeeks.org/fundamentals-of-algorithms/
4. http://www.citc.ui.ac.ir/zemoni/cls.pdf

COURSE EVALUATION SCHEME:

Laboratory Component	**Lab Record**	**30 Marks**
	Lab Test	**20 Marks**
	CIE Marks:	**50 Marks**
	SEE Marks	**50 Marks**

CO-PO MAPPING:

POS / COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	-	-	-	-	-	-	-	-	-	-	1	-	1	-
CO2	-	3	-	-	-	-	-	-	-	-	-	2	2	2	2
CO3	-	3	-	-	-	-	-	-	-	-	-	2	2	2	2
CO4	-	-	3	2	-	-	-	-	-	-	-	2	2	2	2

OOPs with C++					
Course Code	**L:T:P**	**Credits**	**Exam Marks**	**Exam Duration**	**Course Type**
23CST46A	2:0:2	3	CIE:50SEE:50	3 Hours	AEC/ SEC

PREREQUISITES:

Basic understanding of Object Oriented Concepts.

Course Objectives :

The Student will be able to:

- Understand object-oriented programming using C++
- Gain knowledge about the capability to store information together in an object.
- Learn the capability of a class to rely upon another class and functions.
- Create and process data in files using file I/O functions
- Understand the generic programming features of C++ including Exception handling

Syllabus

Module I

An overview of C++: What is object-Oriented Programming? Introducing C++ Classes, The General Form of a C++ Program. Classes and Objects: Classes, Friend Functions, Friend Classes, Inline Functions, Parameterized Constructors, Static Class Members, When Constructors and Destructors are Executed, The Scope Resolution Operator, Passing Objects to functions, Returning Objects, Object.

Ch 11, Ch 12

Module II

Functions Overloading,,Copy Constructors: Functions Overloading, Overloading Constructor Functions. Copy Constructors, Default Function Arguments, Function Overloading and Ambiguity.

Ch 13, Ch 14

Module III

Operator Overloading: Creating a Member Operator Function, Operator Overloading for the operators +,-,<<and >>

Ch 15

Module IV

Inheritance: Base-Class Access Control, Inheritance and Protected Members, Inheriting Multiple Base Classes , Constructors, Destructors and Inheritance, Granting Access, Virtual Base Classes

Ch 16

Module V

Virtual Functions and Polymorphism: Virtual Functions, The Virtual Attribute is Inherited, Virtual Functions are Hierarchical, Pure Virtual Functions, Using Virtual Functions, Early vs Late Binding.

Ch 17

Course outcomes:

The Student will be able to:

- Illustrate the basic concepts of object-oriented programming.
- Apply the concepts class, object, overloading,inheritance and polymorphism for the given real world scenario.
- Analyze the concepts of object oriented programming.
- Design and develop an optimized solution using object oriented concepts.
- Demonstrate the concepts of C++ programming using relavant tool.

Text Books:

1. Herbert schildt, The Complete Reference C++, 4 th edition, TMH, 2005

Reference Books:

1. Balagurusamy E, Object Oriented Programming with C++, Tata McGraw Hill Education Pvt.Ltd., Sixth Edition 2016.
2. Bhave , " Object Oriented Programming With C++", Pearson Education , 2004.
3. A K Sharma , "Object Oriented Programming with C++", Pearson Education, 2014

Assessment Details (both CIE and SEE

E-Resources:

- https://www.youtube.com/watch?v=BClS40yzssA
- https://www.youtube.com/watch?v=p8ehAjZWjP

Practical Component

Sl No	Experiments
1	Develop a C++ program to find the largest of three numbers
2	Develop a C++ program using classes to display student name, roll number, marks obtained in two subjects and total score of student
3	Develop a C++ program for a bank empolyee to print name of the employee, account_no. & balance. Print invalid balance if amount
4	Develop a C++ program to demonstrate function overloading for the following prototypes. add(int a, int b) add(double a, double b)
5	Develop a C++ program to implement Multiple inheritance for performing arithmetic operation of two numbers
6	Develop a C++ program using Constructor in Derived classes to initialize alpha, beta and gamma and display corresponding values.

PO-PSO Mapping:

PO'S	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO1 0	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	-	-	-	-	-	-	-	-	-	-	1	-	-	3
CO2	3	-	-	-	-	-	-	-	-	-	-	-	-	-	3
CO3	-	3	-	-	-	-	-	-	-	-	-	-	-	-	3
CO4	-	-	3	-	-	-	-	-	-	-	-	-	-	-	3
CO5	-	-	-	2	2	-	-	-	-	-	-	-	-	-	3
CO	3	3	3	2	2	-	-	-	-	-	-	1	-	-	3

UI/UX		Semester	4
Course Code	**23CSL46C**	CIE Marks	50
Teaching Hours/Week (L: T:P: S)	1:0:0:0	SEE Marks	50
Total Hours of Pedagogy	14	Total Marks	100
Credits	01	Exam Hours	01
Examination type (SEE)	Theory (MCQ)		

Course objectives:

- Understand user experience design requirements, with design goals, metrics andtargets.
- Explore different prototyping methods, UX design principles with case examples.
- Understand the role of design thinking concepts and mental models in UX design.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) need not to be only traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotes Criticalthinking.
5. Adopt Case study Based Learning (CBL), which fosters students' analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather thansimply recall it.
6. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

Module-1

Introduction: Usability to user experience, Emotional impact as part of user experience, Userexperience needs a business case.
Extracting Interaction Design Requirements: Needs & Requirements, Formal requirement extraction, Methods for requirement extraction.

Module-2

Design Thinking, Ideation, and Sketching: Design Thinking, Design Perspectives, User Personas,Ideation, Sketching.
Mental Models and Conceptual Design: Storyboards, Design influencing user behaviour.

Module-3

Design Production: Detailed Design, Wireframes.
UX Goals, Metrics and Targets: UX Goals, UX Measures, Measurement instruments, UX Metrics.

Module-4

Prototyping: Depth & breadth of a prototype, Fidelity of prototypes, Paper prototypes.
Connections with Software Engineering: Foundations for success in SE-UX development, The challengeof connecting SE and UX.

Module-5

UX Design Guidelines: Using and interpreting design guidelines, Human memory limitations, UX designguidelines & examples, Planning, Translation, Physical action, Outcomes, Assessment, Overall.

Course outcome (Course Skill Set)

At the end of the course the student will be able to:

1. Explain the user experience design requirements.
2. Relate design thinking concepts and mental models to UX design.
3. Illustrate UX design in line with design goals, metrics and targets.
4. Demonstrate different prototyping in relation with software engineering.
5. Explain UX design principles with case examples.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

Continuous internal Examination (CIE)

- For the Assignment component (CCE) of the CIE, there are 25 marks and for the InternalAssessment Test component, there are 25 marks.
- The first test will be administered after 40-50% of the syllabus has been covered, and thesecond test will be administered after 85-90% of the syllabus has been covered
- Any two assessment methods mentioned in the 22OB2.4, if an assessment is project-basedthen only one assessment for the course shall be planned. The teacher should not conduct two assignments at the end of the semester if two assignments are planned.
- For the course, CIE marks will be based on a scaled-down sum of two tests and othermethods of assessment.

Internal Assessment Test question paper is designed to attain the different levels of Bloom'staxonomy as per the outcome defined for the course.

Semester End Examinations (SEE)

SEE paper shall be set for 50 questions, each of the 01 marks. The pattern of the question paper is MCQ (multiple choice questions). The time allotted for SEE is **01 hour.** The student has to secure a minimum of 35% of the maximum marks meant for SEE.

Suggested Learning Resources:

Books

1. REX HARTSON and PARDHA S. PYLA, The UX Book-Process and Guidelines for Ensuring a QualityUser Experience, Morgan Kaufmann, Elsevier, 2012.

Web links and Video Lectures (e-Resources):

- https://www.freecodecamp.org/news/ui-ux-design-tutorial-from-zero-to-hero-with- wireframe-prototype-figma/
- https://www.edureka.co/blog/ui-ux-design-tutorial/
- https://www.udemy.com/course/introtoux/

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- UI design demonstrations covering different UX design principles/concepts (specified in thesyllabus) using UI/UX tools like Lunacy, framer, penpot, visily etc.

V SEMESTER

COURSE SYLLABUS

OPERATING SYSTEMS					
Course Code	**L:T:P:S**	**Credits**	**Exam Marks**	**Exam Duration**	**Course Type**
22CST51	3:0:0:0	3	CIE:50 SEE:50	3 Hours	PCC/BSC

PREREQUISITES:

Good knowledge of C, Computer Organization and Architecture, x86 Assembly level programming.

Course Objectives :

The Student will be able to:

- To Demonstrate the need for OS and different types of OS
- To discuss suitable techniques for management of different resources
- To demonstrate different APIs/Commands related to processor, memory, storage and file system management.

Teaching-Learning Process (General Instructions)

Teachers can use the following strategies to accelerate the attainment of the various course outcomes.

7. Lecturer methods (L) need not to be only traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
8. Use of Video/Animation to explain functioning of various concepts.
9. Encourage collaborative (Group Learning) Learning in the class.
10. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
11. Role-play for process scheduling.
12. Demonstrate the installation of any one Linux OS on VMware/Virtual Box

Syllabus

Module I

Introduction to operating systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments.

Operating System Services: User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System generation; System boot.

Textbook 1: Chapter – 1 (1.1-1.12), 2 (2.2-2.11) 8 Hours

Module II

Process Management: Process concept; Process scheduling; Operations on processes; Inter process communication.

Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues.

Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Thread scheduling; Multiple-processor scheduling.

Textbook 1: Chapter – 3 (3.1-3.4), 4 (4.1-4.4), 5 (5.1 -5.5) 8 Hours

Module III

Process Synchronization: Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores.

Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

Textbook 1: Chapter – 6 (6.1-6.6), 7 (7.1 -7.7) **8 Hours**

Module IV

Memory Management: Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.

Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing.

Textbook 1: Chapter -8 (8.1-8.6), 9 (9.1-9.6) **8 Hours**

Module V

File System, Implementation of File System: File system: File concept; Access methods; Directory and Disk structure; File system mounting; File sharing; **Implementing File system:** File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.

Textbook 1: Chapter – 10 (10.1-10.5) ,11 (11.1-11.5) **8 Hours**

Course outcomes:
The Student will be able to:

CO 1. Explain the structure and functionalities of operating system.

CO 2. Apply appropriate algorithms for the given problem.

CO 3. Analyze the various techniques used in operating system design and implementation.

CO 4. Compare various types of operating system based on different attributes.

CO 5. Prepare a comparative report on functionalities of operating system.

Text Books:

2. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 8th edition, Wiley-India, 2015

Reference Books:

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

E-Resources:

5. https://youtu.be/mXw9ruZaxzQ
6. https://youtu.be/vBURTt97EkA
7. https://www.tutorialspoint.com/operating_system/index.htm.
8. https://www.studytonight.com/operating-system/.

CO-PO-PSO Mapping

PO'S	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO1 0	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO2	3	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO3	-	3	-	-	-	-	-	-	-	-	-	-	3	-	-
CO4	-	-	2	-	-	-	-	-	-	-	-	-	3	-	-
CO5	-	-	-	-	-	-	-	-	-	2	-	2	3	-	-
CO	3	3	2							2		2	3	-	-

Theory of Computation

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
22CST53	3:0:0:0	3	CIE: 50 SEE: 50	3 Hours	PCC

Description of the course: This course introduces the theory of computation through a set of abstract machines that serve as models for computation - finite automata, pushdown automata, and Turing machines - and examines the relationship between these automata and formal languages. In this course we will introduce various models of computation and study their power and limitations. We will also explore the properties of the corresponding language classes defined by these models and the relations between them.

Prerequisite: Discrete Mathematics, Design and Analysis of Algorithms.

Course Objectives:

This course will enable a student to:

- Study abstract computing models.
- Formalization of the notion of problems via formal languages.
- Learn Grammar and Turing Machine.
- Learn about the theory of computability and complexity.
- Understand a hierarchy of classes of problems or formal languages.

Syllabus

Module – I

Finite Automata: The Central Concepts of Automata Theory, An Informal Picture of Finite Automata, Deterministic Finite Automata (DFA), Nondeterministic Finite Automata (NFA), Finite Automata with Epsilon Transitions (ε-NFA). **08 Hours**

Module – II

Regular Expressions and Languages: Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, NFA to DFA, RE to DFA, DFA to RE Conversions: State/loop elimination, Properties of Regular Languages: Pumping Lemma for Regular languages, Closure properties. **08 Hours**

Module – III

Context Free Grammars (CFG) and Languages: Context Free Grammars: Definition, Derivations using a grammar, Leftmost and Rightmost derivations, Language of grammar, Sentential form, Parse trees, Ambiguity in grammars and Languages: Ambiguous Grammar, Removing ambiguity from Grammars. **08 Hours**

Module – IV

Properties of CFL: Eliminating useless symbols, unit productions, useless production, and Є-productions, Normal Forms: Chomsky Normal Form.

Pushdown Automata: Definition of the Pushdown Automatan, Languages of a PDA, Equivalence of PDA's and CFG's, Deterministic Pushdown Automata. **08 Hours**

Module – V

Turing Machines: Problems that Computers cannot solve; The Turing Machine; Programming techniques for Turing Machines; Extensions to the basic Turing Machines; Turing Machine and Computers. **08 Hours**

Course Outcomes:

On completion of the course, student will be able to:

- Acquire fundamental understanding of the core concepts in automata theory and Theory of Computation.
- Learn how to translate between different models of Computation (e.g., Deterministic and Non-deterministic and Software models).
- Design Grammars and Automata (recognizers) for different language classes and become knowledgeable about restricted models of Computation (Regular, Context Free) and their relative powers.
- Develop skills in formal reasoning and reduction of a problem to a formal model, with an emphasis on semantic precision and conciseness.

Textbook:

1. John E Hopcroft, Rajeev Motwani, Jeffery D Ullman, "Introduction to Automata Theory, Languages and Computation", (Chapter No.: 1.5, 2, 3, 4.1, 4.2, 5, 6, 7.1, 8), 3rd Edition, Pearson Education, 2013, ISBN-13: 978-8131720479.

Reference Books:

1. Peter Linz, "An introduction to Formal languages and Automata", 5th Edition, Cathleen Sether Publishers, 2012, ISBN-13: 9781449615529.
2. Michael Sipser: "Introduction to the Theory of Computation", 3rd Edition, Cengage learning, 2013.

E-Resources:

1. https://plato.stanford.edu/entries/computational-complexity/#TecDev
2. https://www.cse.iitm.ac.in/~shwetaag/col705.html
3. https://www.cs.ucy.ac.cy/~mavronic/Classes/cs211/index.html
4. https://www.cse.csusb.edu/egomez/cs601.html
5. https://www-e.openu.ac.il/courses/20585.html

COURSE EVALUATION SCHEME:

Theory Component	IAT-1 after completion 45 to 50% Syllabus	25 Marks
	IAT-2 after completion 95 to 100% Syllabus	25 Marks
	Sum of two IATs	**50 Marks**
	Total 25 Marks : Reduced to 25 Marks	
	CCE-1	25 Marks
	CCE-2	25 Marks
	Sum of two CCEs	**50 Marks**
	Total 25 Marks : Reduced to 25 Marks	
Grand Total of CIE Marks		**50 Marks**
SEE Marks		**50 Marks**
Total Marks (CIE + SEE)		**100 Marks**

CO-PO MAPPING:

POS / COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	-	-	-	-	-	-	-	-	-	-	2	-	2	-
CO2	-	2	-	-	-	-	-	-	-	-	-	2	2	2	2
CO3	-	3	-	-	-	-	-	-	-	-	-	2	2	2	2
CO4	-	-	3	-	-	-	-	-	-	-	-	2	2	2	2

DISTRIBUTED SYSTEMS

SEMESTER – VI

Course Code	Course Type	L:T:P:S	CIE Marks	SEE Marks	Total Marks	Exam Hours
22CST541	Theory	3:0:0:0	50	50	100	03
Total Hours of Pedagogy		40			Credits	03

Course objectives: This course will enable students to

- Explain distributed system, their characteristics, challenges and system models.
- Describe IPC mechanisms to communicate between distributed objects
- Illustrate the operating system support and File Service architecture in a distributed system
- Analyze the fundamental concepts, algorithms related to synchronization.
- Learn various distributed transaction mechanisms.

Module – 1	**Teaching Hours**
Characterization of Distributed Systems: Introduction, Examples of DS, Resource sharing and the Web, Challenges **System Models:** Architectural Models, Fundamental Models	**8 Hours**
Module – 2	
Inter Process Communication: Introduction, API for Internet Protocols, External Data Representation and Marshalling, Client – Server Communication, Group Communication **Distributed Objects and RMI:** Introduction, Communication between Distributed Objects, RPC, Events and Notifications	**8 Hours**
Module – 3	
Operating System Support: Introduction, The OS layer, Protection, Processes and Threads, Communication and Invocation , Operating system architecture **Distributed File Systems:** Introduction, File Service architecture, Sun Network File System	**8 Hours**
Module – 4	
Time and Global States: Introduction, Clocks, events and process status, Synchronizing physical clocks, Logical time and logical clocks, Global states **Coordination and Agreement:** Introduction, Distributed mutual exclusion, Elections	**8 Hours**
Module – 5	
Distributed Transactions: Introduction, Flat and nested distributed transactions, Atomic commit protocols, Concurrency control in distributed transactions, distributed deadlocks	**8 Hours**

Course outcomes: The students should be able to:

- Explain the characteristics of a distributed system along with its and design challenges
- Illustrate the mechanism of IPC between distributed objects
- Describe the distributed file service architecture and the important characteristics of SUN NFS.
- Discuss concurrency control algorithms applied in distributed transactions
- Apply the knowledge of distributed transactions and interpret its drawbacks.

Text Books:
1. George Coulouris, Jean Dollimore and Tim Kindberg: Distributed Systems – Concepts and Design, 5th Edition, Pearson Publications, 2009
Reference Books:
1. Andrew S Tanenbaum: Distributed Operating Systems, 3rd edition, Pearson publication, 2007 2. Ajay D. Kshemkalyani and Mukesh Singhal, Distributed Computing: Principles, Algorithms and Systems, Cambridge University Press, 2008 3. Sunita Mahajan, Seema Shan, " Distributed Computing", Oxford University Press,2015

Course Name: Computer Graphics

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
22CST544	**3:0:0:0**	**03**	**CIE:50 SEE:50**	**03 Hours**	**PEC**

Course Objectives:

This course will enable students to:

- Understand the Fundamentals of Computer Graphics.
- Apply Object Representation Techniques and Basic Transformations.
- Implement and Evaluate Lighting Models, Shading Techniques, and Intensity Representation.
- Understand and Utilize Color Models, Texture Synthesis, and 3D Viewing Techniques.
- Develop Skills in Rendering Techniques and Graphics Hardware/Software.

Syllabus

MODULE- I

Overview of Computer Graphics: Historical Development of the Field, Major Issues and Concerns in Computer Graphics, Preliminaries: Basics of Graphics System, Graphics Pipeline: Stages of Rendering Process, Role of Graphics Libraries

Object Representation Techniques: Categorization of Representation Techniques, Boundary Representation Techniques, Spline Representations, Space-partitioning Representation, Other Representations, Issues in Model Selection.

08 Hours

MODULE - II

Modeling Transformations: Basic Transformations, Matrix Representation and Homogeneous Coordinate System, Composition of Transformations, Transformations in 3D,

Illumination, Lighting Models, and Intensity Representation: Background, Simple Lighting Model, Shading Models, Handling the Shadow Effect, Intensity Representation

08 Hours

MODULE - III

Color Models and Texture Synthesis: Physiology of Vision, Color Models: RGB Color Model, XYZ Color Model, CMY Color Model, HSV Color Model, Texture Synthesis: Projected Texture, Texture Mapping, Solid Texture.

3D Viewing: 3D Viewing Transformation: Setting up a View Coordinate System, Viewing Transformation, Projection: Types of Projections, Projection Transformation, Canonical View Volume and Depth Preservation, Window-to-viewport Transformation

08 Hours

MODULE- IV

Rendering: Scan Conversion of a Line Segment, Circle Scan Conversion, Fill Area Scan Conversion, Character Rendering, Anti-aliasing.

Graphics Hardware and Software: Generic Architecture, Input and Output of Graphics System, GPU and Shader Programming, Graphics Software and OpenGL

08 Hours

MODULE- V:

Computer Animation: Traditional Animation Techniques, Principles of Animation, Timing, Action Planning and Layout, Animation Techniques, Keyframing: Character and Facial Animation, Deformation, Motion Capture, Physically based Methods and Procedural Techniques.

Multimedia and Hypermedia: Hypermedia, Multimedia Authoring, Components of Multimedia: Basics of Digital Audio, Digital Video Fundamentals, Data Compression Standards: JPEG Image Compression Standard, H.261 Digital Video Compression Standard, MPEG Standard

08 Hours

Text Books:

1. Computer Graphics by Samit Bhattacharya, Oxford University Press, 2015

Reference books:

1. Computer Graphics by John F H, Andries et.al, Third Edition, Addison-Wesley, 2014

COURSE OUTCOMES:

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

CO1	Describe the Historical Evolution and Fundamental Concepts of Computer Graphics
CO2	Apply Object Representation Techniques and Basic Transformations in 2D and 3D Graphics.
CO3	Implement and Evaluate Various Lighting Models, Shading Techniques, and Intensity Representation Methods
CO4	Utilize Color Models, Texture Synthesis Techniques, and 3D Viewing Methods for Effective Graphics Design
CO5	Demonstrate Proficiency in Rendering Techniques and Graphics Hardware/Software Tools

CO-PO MAPPING:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3
C13.1	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.2	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.3	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.4	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-
C13.5	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-

Course Name: WEB TECHNOLOGY LAB

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
22CSL55	0: 0 : 2	1	CIE:50 SEE:50	3 Hours	PCCL

Sl.No.	List of Programs
1.	Write and Implement HTML script to display employee details like name, address, mobile number, email id etc similar to a telephone directory.
2.	Demonstrate a HTML program to display a nested list to list down all the elements serviced by an event management company. The list should be a nested list with main events and subevents.
3.	Implement a HTML and CSS script to create a webpage with table structure containing alternative backgrounds using class selector functionalities.
4.	Design a HTML and CSS program for the coverpage which displays the events taking place in and around the state.
5.	Construct a HTML and javascript program to implement a simple banking application using SQL database. The application should provide features like withdraw, deposit, balance enquiry etc.
6.	Implement a HTML and javascript program to create a registration page having fields name, username, email Id, password & re-enter password and apply validation using match & equal functions.
7.	Create a webpage to fetch the details of the event and display the invitation using HTML and jQuery.
8.	Design a webpage to accept event organizer name from the user and display it on the webpage using HTML and jQuery
9.	Illustrate a HTML and bootstrap program to display glyphicons like envelop, print, search etc. Also, create buttons having glyphicons as links to carry out specific tasks.
10.	Develop a XML program to store book details like title, author, and publication year, price etc. using RSS.

Sl. No.	Course Outcomes
CO1	Analyze the web layouts with style sheets and web screens in a presentable form.
CO2	Create interactive web pages through form validations.
CO3	Design scripts using JavaScript in a web page. .
CO4	Integrate responsive webpages for frameworks.
CO5	Develop applications by using synchronous and asynchronous communication over web.

Text Books:

1. Robert W. Sebesta: “Programming the World Wide Web”, Pearson, 4th Edition, 2012, ISBN: 978-81-317-6458-9.

2. Jon Duckett: “Web Design with HTML, CSS, JavaScript and jQuery Set”, Wiley, 1st Edition, 2014, ISBN 13: 978-1118907443.

3. Silvio Moreto, Matt Lambert, Benjamin Jakobus, Jason Marah:” Bootstrap 4 - Responsive Web Design”, Packt Publishing, 2016, ISBN 978-1-78839-731-5

Reference Books:

1. Jake Spurlock: "Bootstrap, Shroff", O"Reilly Media, United States of America, 1st Edition, 2013, ISBN: 978 -1 -4493-4391-0.

2. Bear Bibeault, Yehuda Katz and Aurelio De Rosa: "jQuery in Action", Dreamtech Press, New Delhi, India, 3rd Edition, 2015, ISBN: 978-1617292071

E-Resources:

1. http://www.w3schools.com/
2. https://www.tutorialspoint.com/
3. https://www.javascript.com/learn/
4. https://learn.jquery.com

CO-PO & PSO Mapping:

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO's		
													PSO1	PSO2	PSO3
C212.1	-	-	2	-	-	-	-	-	-	-	-	-	2	-	-
C212.2	-	-	2	-	-	-	-	-	-	-	-	-	2	-	-
C212.3	3	-	-	-	-	-	-	-	-	-	-	-	2	-	-
C212.4	-	-	2	-	-	-	-	-	-	-	-	-	2	-	-
C212.5	-	-	2	-	-	-	-	-	-	-	-	-	2	-	-
C212	3	-	2	-	-	-	-	-	-	-	-	-	2	-	-

COURSE EVALUATION SCHEME:

Evaluation Type		Component	Max Marks	Marks Reduced	Min Marks	Evaluation Details
Laboratory Component	Lab Test	Lab Test-I	15	--	**12**	40% of the sum of two tests
		Lab Test-II	15			
	Lab Record	-	20	--	**8**	40% of the Lab record
	Total CIE- Practical's		**50**	--	**20**	40% of the Lab test and record
IAT in Practical Total Marks: 50 Marks. The minimum passing mark for the IAT to appear for SEE Practica is 40% of the maximum marks (20 marks out of 50(Lab CIE))						
SEE Practical Exam is 50 Marks The minimum passing mark for the SEE practical is 40% of the maximum marks (20 marks out of 50)						

MINI PROJECT

Course Code	L:T:P	Credits	Total Marks	Exam Hours	Course Type
22CSP57	**0:0:4**	**02**	**CIE:100**	**03**	**Project**

Course Objectives:

1. Motive students to work on literature survey, to understand the domain of their interest preferable in advanced and emerging technologies.
2. Summarize the literature survey and finalize their project work under selected domain.
3. Develop team work & presentation skills, and prepare the report.

Course Content:

- Survey and study of published literature on the assigned topic related to emerging technologies like machine learning, Cyber security, Data Science, Web technologies etc.
- Working out a preliminary Approach to the Problem relating to the assigned topic.
- Preparing a written report on the Study conducted for presentation to the Department.
- Final Seminar, as oral Presentation before a Departmental Committee.

Assessment Methods

Parameter/Rubrics	Marks
Formation of team, selection of domain and submission of Synopsis	20
Carry out literature survey	24
Design and development, Preparation of the report	36
Rubrics based evaluation of Presentation and Viva	20
CIE Total Marks	**100**

Course Outcomes

Sl. No.	Course Outcomes
CO1	Review the current state of Art and trends in their area of interest in current technologies and identify a suitable problem in their chosen subject domain with justification.
CO2	Survey the available research literature/documents for the tools and techniques to be used.
CO3	Examine the functional, non-functional, and performance requirements of their chosen problem definition.
CO4	Design system architecture and different components and develop all the system components using appropriate tools and techniques.
CO5	Work effectively in a team and use good project management practices and defend the project work as a team.

RESEARCH METHODOLOGY &IPR

Course Code	22CST58	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	2:2:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	03	Exam Hours	03

Prerequisites: Literature survey, Requirement analysis

Course objectives:

1. To give an overview of the research methodology and explain the technique of defining a research problem
2. To explain the functions of the literature review in research.
3. To explain carrying out a literature search, its review, developing theoretical and conceptual frame works and writing a review and research reports.
4. To explain various forms of the intellectual property, its relevance and business impact in the changing global business environment.
5. To discuss leading International Instruments concerning Intellectual Property Rights.
6.

Teaching-Learning Process (General Instructions)

These are sample Strategies; which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) need not to be only a traditional lecture method, but alternativeeffective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher order Thinking) questions in the class, which promotescritical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students‘ Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
6. Introduce Topics in manifold representations.
7. Show the different ways to solve the same problem with different circuits/logic andencourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it Helps improve the students' understanding.

Syllabus

Module – I

Research Methodology: Introduction, Meaning of Research, Objectives of Research, Types of Research, Research Approaches, Significance of Research, Research Methods versus Methodology, Research and Scientific Method, Research Process, Criteria of Good Research, Problems Encountered by Researchers in India.

Defining the Research Problem: Research Problem, Selecting the Problem, Necessity of Defining the Problem, Technique Involved in Defining a Problem, An Illustration.

Text Book 1 - Chapter 1, 2 **08 Hours**

Module – II

Reviewing the literature: Place of the literature review in research, bringing clarity and focus to research problem, improving research methodology, broadening knowledge base in research area, enabling contextual findings, Review of the literature, searching the existing literature, reviewing the selected literature, developing a theoretical framework, developing a conceptual framework, writing about the literature reviewed.

Text Book 4 - Chapter 3.

Research Design: Meaning of Research Design, Need for Research Design, Features of a Good Design, Important Concepts Relating to Research Design, Different Research Designs.

Text Book 1 - Chapter 3. **08 Hours**

Module – III

Data Collection: Introduction, Experimental and Surveys, Collection of Primary Data, Collection of Secondary Data, Selection of Appropriate Method for Data Collection, Case Study Method.

Text Book 1 - Chapter 6.

Ethics in Engineering Research- Ethics in Engineering Research Practice, Types of Research Misconduct, Ethical Issues Related to Authorship.

Text Book 2 - Chapter 5. **08 Hours**

Module – IV

Interpretation and Report Writing- Meaning of Interpretation, Techniques of Interpretation, Precautions in Interpretation, Significance of Report writing, Different steps in writing report, Layout of the research report, Types of reports, Oral presentation, Mechanics of writing a research report, Precautions for writing research reports, Conclusion.

Text Book 1 - Chapter 19.

Technical Writing and Publishing - Free Writing and Mining for Ideas, Attributes andReasons of Technical Writing, Patent or Technical Paper —The Choice, Writing Strategies, Journal Paper: Structure and Approach, Language Skills, Writing Style, and Editing, Rules of Mathematical Writing, Publish Articles to Get Cited, or Perish.

Text Book 2 - Chapter 6.

Communicating Research Work: Presentation Skills - Oral Presentations - Language Choices, Delivery, Poster Presentations, Presentation Preparation Guidelines.

Text Book 2 - Chapter 9. **08 Hours**

Module – V

Intellectual property: an introduction - Intellectual property types, More patent basics.
Text Book 3 - Module 1 - 1, 2.
Patents- Detailed overview of patents-What is a patent, What can be the subject of a patent, Why are patents important. Legal requirements for patentability - Novelty, Inventive step/non obviousness, Industrial application/utility, Patentable subject matter, Disclosure requirement.
Text Book 3 - Module 2 - 1.1, 1.2, 1.3, 2.1, 2.2, 2.3, 2.4, 2.5
Patent application preparation - Preparing patent applications - Obtaining invention disclosures from Inventors, Identifying patentable inventions, Understanding the invention(core inventive concept), Inventorship. Typical parts of the patent Application - Request, Description, Claims, Drawings, Abstract, and Application format.
Text Book 3 - Module 3 - 1.1, 1.2, 1.3, 1.4, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6. 08 Hours

Teaching-Learning Process forall modules

Chalk and board, Active Learning, PPT Based presentation, Video

Course Outcomes (Course Skill Set)

At the end of the course the student will be able to:

CO1. Explain the significance of engineering research.

CO2. Explore the procedure of Literature Review and Technical Reading.

CO3. Elucidate the diverse methodologies of data collection and the ethical consideration within research

CO4. Understand the methodologies employed in report writing, technical writing, and presentation skills.

CO5. Elaborate on the basics of patent law and the procedure for drafting patents

Assessment Details (both CIE and SEE)

Component		Weightage (%)	
CIE's	CIE 1 5th week	20	60
	CIE 2 10th week	20	
	CIE 3 15th week	20	
AAT's	AAT-1 10th week	10	
	AAT-2	10	
	AAT-3	20	
Continuous Internal Evaluation Total Marks: 100. Reduced to 50 Marks			
Semester End Examination (SEE) Total Marks: 100. Reduced to 50 Marks			

Textbooks

1	Research Methodology: Methods and Techniques	C. R. Kothari, Gaurav Garg	New Age Internat ional	4th Edition, 2019
2	Engineering Research Methodology: A Practical Insight for Researchers	Dipankar Deb, Rajeeb Dey, Valentina E. Balas	Intelligent Systems Reference Library	1st Edition, 2019
3	WIPO (2022), WIPO Patent Drafting Manual, 2nd edition. Geneva: WIPO.	DOI: 10.34667/tind.44657 ISBN: 978-92-805-3264-7	World Intellectual Property Organization	Second edition
4	RESEARCH METHODOLOGY a step-by-step guide for beginners.	Ranjit Kumar	SAGE Publications India Pvt Ltd.	3rd Edition, 2011.

Reference Books				
1	"Research Methods for Engineers"	David V. Thiel	Cambridge University Press	2020

Online Resources

1. https://onlinecourses.nptel.ac.in/noc22_ge08/preview
2. https://archive.nptel.ac.in/courses/127/106/127106227/
3. https://onlinecourses.swayam2.ac.in/cec20_hs17/preview
4. https://archive.nptel.ac.in/courses/110/105/110105139/
5. www.indiacode.nic.in

VI SEMESTER

COURSE SYLLABUS

CLOUD COMPUTING

Course Code	L:T:P	Credits	Exam Marks	Exam Duration	Course Type
21CST61	3:0:0:0	03	CIE:50 SEE:50	03 Hours	PCC

Course Objectives:

This course will enable students to:

- Learn the basic concepts of cloud computing, characteristics, benefits, types, and historical developments.
- Understand virtualization, including its importance, software operation, hypervisors, and the workings of virtual machines.
- Understand Amazon Web Services (AWS), covering service categories, global infrastructure, cloud economics, security, and account management.
- Learn about AWS compute services and networking basics for building scalable and secure applications in the cloud.
- Explore AWS storage services and databases for efficient data storage, management, and retrieval.

Module – I

Introduction to Cloud Computing: The Vision of Cloud Computing, Defining a Cloud, A Closer Look, The cloud computing reference model, On Premise Data Center v/s cloud, Characteristics and Benefits, Types of Cloud Computing, Challenges ahead, Historical Developments

08 Hours

Module – II

Virtualization: Describing Virtualization, Understanding the Importance of Virtualization, Understanding Virtualization Software Operation, Understanding Hypervisors, Describing a hypervisor, Understanding the Role of a Hypervisor, Comparing Today's Hypervisors, Understanding How a Virtual Machine Works

08 Hours

Module – III

Introduction to Amazon Web Services (AWS): AWS service and service category overview, AWS Global Infrastructure, AWS Cloud Adoption Framework (AWS CAF), Cloud Economics and Billing, AWS Organizations, AWS Identity and Access Management (IAM),. AWS shared responsibility model, Securing a new AWS account, Securing data on AWS.

08 Hours

Module – IV

Compute services overview: Amazon EC2, Amazon EC2 cost optimization, Container services, Introduction to AWS Lambda, Introduction to AWS Elastic Beanstalk.

Networking and Content Delivery: Networking basics, Amazon Virtual Private Cloud (Amazon VPC), VPC networking, VPC security, Amazon Route 53, Amazon CloudFront

08 Hours

Module – V

Storage: Amazon Simple Storage Service (Amazon S3), Amazon Elastic Block Store (Amazon EBS), Amazon Elastic File System (Amazon EFS), Amazon Simple Storage Service Glacier.

Databases: Amazon Relational Database Service (Amazon RDS), Amazon DynamoDB, Amazon Redshift, Amazon Aurora.

08 Hours

LAB Experiments:

1. Install and configure a virtual machine using virtualization software (such as VirtualBox or VMware).
2. Set up a new AWS account and implement basic security measures such as Multi-Factor Authentication (MFA), IAM roles, and policies.
3. Create IAM users and groups with specific permissions. Assign policies to the groups and demonstrate how to restrict and allow access to different AWS services.
4. Secure a new AWS account by enabling Multi-Factor Authentication (MFA) for the root account and setting up a billing alert.
5. Launch an EC2 instance, connect to it using SSH, and install a web server (e.g., Apache or Nginx). Configure the security group to allow HTTP and SSH access, and deploy a simple HTML page.
6. Use AWS Cost Explorer to analyze the cost of your EC2 instances. Apply cost-saving techniques such as right-sizing instances, utilizing reserved instances, and setting up a schedule for turning off non-essential instances. Document the cost savings achieved.
7. Create an Amazon S3 bucket and configure it with versioning and lifecycle policies. Set up Amazon Elastic File System (EFS) and mount it to an EC2 instance. Compare the use cases and performance of S3 and EFS for different types of data storage
8. Design and configure a Virtual Private Cloud (VPC) with public and private subnets, route tables, and a NAT gateway. Launch instances in both subnets and demonstrate connectivity between them.
9. Create a simple AWS Lambda function that processes data (e.g., a function that converts text to uppercase). Trigger the function using AWS API Gateway and test it with different inputs.
10. Create a MySQL database on Amazon RDS, connect using a MySQL client, and perform basic SQL operations such as creating a table, inserting data, and querying the database.

Course Outcomes:

On completion of this course, students should be able to:

CO1: Understand cloud computing, cloud characteristics, types of cloud, benefits and challenges.

CO2: Analyze virtualization, hypervisors, their role, operation of virtual machines and providing a solid foundation for leveraging virtualization technologies.

CO3: Apply the knowledge of AWS service to secure the cloud data.

CO4: Apply the knowledge of AWS to launch EC2 instance and secure the cloud using VPC.

CO5: Evaluate different AWS storage services and database service and compare them each.

Text Books:

1. Buyya, Rajkumar, James Broberg, and Andrzej M. Goscinski, eds. Cloud computing: Principles and paradigms. Vol. 87. John Wiley & Sons, 2011, ISBN: 978-0-470-88799-8.
2. AWS Certified Solutions Architect Official Study Guide: Associate Exam by Joe Baron, Hisham Baz, Tim Bixler, Biff Gaut, Kevin E. Kelly, Sean Senior, John Stamper- ISBN10 - 1119138558
3. Velte T, Velte A, Elsenpeter R. Cloud computing, a practical approach. McGraw-Hill, Inc.; Sep 22. 2009, ISBN: 9780070683518.

Reference Books:

1. Thomas Erl: “Cloud Computing”, Pearson Education, 1st Edition, 2014, ISBN-13: 978-9332535923.
2. Judith Hurwitz, Marcia Kaufman, Fern Halper: “Cloud Computing for dummies”, Wiley,1st Edition, 2009, ISBN: 9780470484708.
3. Velte, Anthony T., Toby J. Velte, and Robert Elsenpeter. "Cloud Computing: A Practical Approach." (2009), ISBN: 9780071626941

E-Resources:

1. http://www.buyya.com/MasteringClouds/ToC-Preface-TMH.pdf

Course Name: Digital Image Processing

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
22CST632	1:0:0:0	01	CIE:50 SEE:50	03 Hours	PEC

Course Objectives:

This course will enable students to:

- Comprehend Digital Image Processing Concepts and Applications.
- Master Digital Image Fundamentals and Basic Mathematical Tools.
- Apply Intensity Transformations, Spatial Filtering, and Frequency Domain Techniques.
- Understand and Implement Image Restoration, Reconstruction, and Color Processing Techniques.
- Explore Advanced Image Processing Techniques: Transforms, Morphology, Segmentation, and Feature Extraction.

Syllabus

MODULE- I

Introduction: What is Digital Image Processing?, The Origins of Digital Image Processing, Examples of Fields that Use Digital Image Processing, Fundamental Steps in Digital Image Processing, Components of an Image Processing System

Digital Image Fundamentals: Elements of Visual Perception, Light and the Electromagnetic Spectrum, Image Sensing and Acquisition, Image Sampling and Quantization, Some Basic Relationships Between Pixels, Introduction to the Basic Mathematical Tools Used in Digital Image Processing.

08 Hours

MODULE - II

Intensity Transformations and Spatial Filtering: Background, Some Basic Intensity Transformation Functions, Histogram Processing, Fundamentals of Spatial Filtering, Smoothing (Lowpass) Spatial Filters, Sharpening (Highpass) Spatial Filters, Highpass, Bandreject, and Bandpass Filters from Lowpass Filters, Combining Spatial Enhancement Methods.

Filtering in the Frequency Domain: Background, Preliminary Concepts, Sampling and the Fourier Transform of Sampled Functions, The Discrete Fourier Transform of One Variable, Extensions to Functions of Two

Variables, Some Properties of the 2-D DFT and IDFT, The Basics of Filtering in the Frequency Domain, Image Smoothing Using Lowpass Frequency Domain Filters, Image Sharpening Using Highpass Filters, Selective Filtering, The Fast Fourier Transform

08 Hours

MODULE - III

Image Restoration and Reconstruction: A Model of the Image Degradation/Restoration process, Noise Models, Restoration in the Presence of Noise Only—Spatial Filtering, Periodic Noise Reduction Using Frequency Domain Filtering, Linear, Position-Invariant Degradations, Estimating the Degradation Function, Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering, Constrained Least Squares Filtering, Geometric Mean Filter, Image Reconstruction from Projections.

Color Image Processing: Color Fundamentals, Color Models, Pseudocolor Image Processing, Basics of Full-Color Image Processing, Color Transformations, Color Image Smoothing and Sharpening, Using Color in Image Segmentation, Noise in Color Images, Color Image Compression

08 Hours

MODULE- IV

Wavelet and Other Image Transforms: Preliminaries, Matrix-based Transforms, Correlation, Basis Functions in the Time-Frequency Plane, Basis Images, Fourier-Related Transforms, Walsh-Hadamard Transforms, Slant Transform, Haar Transform, Wavelet Transforms.

Morphological Image Processing: Preliminaries, Erosion and Dilation, Opening and Closing, The Hit-or-Miss Transform, Some Basic Morphological Algorithms, Morphological Reconstruction, Summary of Morphological Operations on Binary Images, Grayscale Morphology.

08 Hours

MODULE- V:

Image Segmentation: Fundamentals, Point, Line, and Edge Detection, Thresholding, Segmentation by Region Growing and by Region Splitting and Merging, Region Segmentation Using Clustering and Superpixels, Region Segmentation Using Graph Cuts, Segmentation Using Morphological Watersheds, The Use of Motion in Segmentation.

Feature Extraction: Background, Boundary Pre-processing, Boundary Feature Descriptors, Region Feature Descriptors, Principal Components as Feature Descriptors, Whole-Image Features, Scale-Invariant Feature Transform (SIFT)

08 Hours

Text Books:

2. Digital Image Processing by Rafael, C Gonzalez, Richard E Woods, Fourth Edition, Pearson 2018

Reference books:

2. Fundamentals of Image Processing by Anil K Jain, Prentice-Hall Inc,

COURSE OUTCOMES:

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

CO1	Understand and Explain Digital Image Processing Concepts and Their Applications
CO2	Apply Mathematical Foundations to Digital Image Processing Tasks
CO3	Perform Image Enhancement Techniques Using Spatial and Frequency Domain Methods
CO4	Implement Advanced Image Processing Techniques for Restoration, Reconstruction, and Color Image Analysis
CO5	Execute Advanced Techniques for Image Segmentation and Feature Extraction

CO-PO MAPPING:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3
C13.1	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.2	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.3	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.4	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-
C13.5	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-

Compiler Design

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
22CST633	3:0:0:0	3	CIE: 50 SEE: 50	3 Hours	PCC

Description of the course: The goal of the course is to provide an introduction to the system software like assemblers, compilers, and macros. It provides the complete description about inner working of a compiler. This course focuses mainly on the design of compilers and optimization techniques. It also includes the design of Compiler writing tools. This course also aims to convey the language specifications, use of regular expressions and context free grammars behind the design of compiler.

Prerequisite: Discrete Mathematics, Design and Analysis of Algorithms, Theory of Computation.

Course Objectives:

This course will enable a student to:

- Understand the fundamentals of compiler design.
- Learn about context-free grammars, parsing algorithms.
- Learn about different forms of intermediate code and techniques for generating efficient and optimized code.
- Gain the Knowledge on concepts of SDD and SDT.
- Explore about code generation strategies, techniques for code optimization.

Syllabus

Module – I

Introduction: Language processors, the structure of a Compiler; The science of building a Compiler; Applications of compiler technology.
Lexical analysis: The Role of Lexical Analyser; Input Buffering; Specifications of Tokens; Recognition of Tokens. **08 Hours**

Module – II

Syntax Analysis-1: Role of Parser; Syntax Error Handling, Error recovery strategies, Context-free Grammars; Writing a Grammar, Top-down Parsing. **08 Hours**

Module – III

Syntax Analysis-2: Top-down Parsing, Bottom-up Parsing, Handle, Handle Pruning, Shift Reduce Parser, Conflicts during shift reduce parser. **08 Hours**

Module – IV

Syntax-Directed Translation: Syntax-directed definitions; Evaluation orders for SDDs; Applications of syntax-directed translation; Syntax-directed translation schemes. **08 Hours**

Module – V

Intermediate Code Generation, Code Optimization: Variants of syntax trees, Three-address code; Issues in the design of Code Generator; The Target Language; Addresses in the target code; Basic blocks and Flow graphs; Optimization of basic blocks; A Simple Code Generator.

08 Hours

Course Outcomes:

On completion of the course, student will be able to:

- Demonstrate Context free grammar and solve the problems of parser.
- Design and implement LL and LR parsers using Top-down and Bottom-up approaches.
- Develop program to solve complex problems in syntax directed translations, syntax directed definitions and to generate intermediate code.
- Apply the target machine's instruction set for code generation and techniques used for code optimization.

Text Book:

3. Alfred V Aho, Monica S.Lam, Ravi Sethi, Jeffrey D Ullman, "Compilers- Principles, Techniques and Tools", 2nd Edition, Pearson Education India, 2013, ISBN 13: 9781530051120. (Chapters 1, 3.1 to 3.4, 4 excluding 4.7.5 and 4.7.6, 5.1 to 5.4, 6.1, 6.2, 6.4, 6.6, 6.7 to 6.9, 7.1 to 7.5, 8.1 to 8.6)

Reference Book:

1. Kaushal Kishor Rastogi, "Compiler Design: Principles, Techniques and Tools", Raj Publications, 2014, ISBN 13: 9788182206267.

E-Resources:

6. https://www.tutorialspoint.com/compiler_design/compiler_design_quick_guide.htm
7. https://www3.nd.edu/~dthain/compilerbook/compilerbook.pdf

COURSE EVALUATION SCHEME:

Theory Component	IAT-1 after completion 45 to 50% Syllabus	25 Marks
	IAT-2 after completion 95 to 100% Syllabus	25 Marks
	Sum of two IATs	**50 Marks**
	Total 25 Marks : Reduced to 25 Marks	
	CCE-1	25 Marks
	CCE-2	25 Marks
	Sum of two CCEs	**50 Marks**
	Total 25 Marks : Reduced to 25 Marks	
Grand Total of CIE Marks		**50 Marks**
SEE Marks		**50 Marks**
Total Marks (CIE + SEE)		**100 Marks**

CO-PO MAPPING:

POS / COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	-	-	-	-	-	-	-	-	-	-	2	-	2	-
CO2	-	2	-	-	-	-	-	-	-	-	-	2	2	2	2
CO3	-	3	-	-	-	-	-	-	-	-	-	2	2	2	2
CO4	-	-	3	-	-	-	-	-	-	-	-	2	2	2	2

Course Name: Soft Computing

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
22CST634	3:0:0:0	03	CIE:50 SEE:50	03 Hours	PEC

Course Objectives:

This course will enable students to:

- Understand the fundamental differences between soft computing and hard computing techniques.
- Explain the basic concepts and applications of fuzzy logic and fuzzy sets.
- Describe the principles and processes involved in evolutionary computing.
- Analyze the working principles and applications of genetic algorithms.
- Explore the concepts and applications of swarm intelligence, including PSO and ACO.

Syllabus

MODULE- I

Introduction:

Introduction, soft computing vs. hard computing, various types of soft computing techniques, and applications of soft computing. Basic tools of soft computing – Fuzzy logic, neural network, evolutionary computing. Introduction: Neural networks, application scope of neural networks, fuzzy logic, genetic algorithm, and hybrid systems.

08 Hours

MODULE - II

Fuzzy Sets and Logic:

Basic concepts of fuzzy logic, Fuzzy sets and Crisp sets, Fuzzy set theory and operations, Properties of fuzzy sets, Fuzzy and Crisp relations, Fuzzy to Crisp conversion. Membership functions, interference in fuzzy logic, fuzzy if-then rules, Fuzzy implications and Fuzzy algorithms, Fuzzyfications and Defuzzification's.

08 Hours

MODULE - III

Evolutionary Computing:

Basic Evolutionary Processes, EV: A Simple Evolutionary System, Evolutionary Systems as Problem Solvers, A Historical Perspective, Canonical Evolutionary Algorithms -Evolutionary Programming, Evolution Strategies, A Unified View of Simple EAs- A Common Framework, Population Size

08 Hours

MODULE- IV

Genetic Algorithm:

Basic concepts, working principle, procedures of GA, flow chart of GA, Genetic representations, (encoding) Initialization and selection, Genetic operators, Mutation, Generational Cycle, Traditional algorithm vs genetic algorithm, simple GA, general genetic algorithm, schema theorem, Classification of genetic algorithm, Holland classifier systems, genetic programming, applications of genetic algorithm, Convergence of GA. Applications and advances in GA, Differences and similarities between GA and other traditional method, applications.

08 Hours

MODULE- V:

Swarm Intelligence:

Swarm intelligence, Particle Swarm Optimization (PSO) Algorithm Formulations, Pseudo-code, parameters, premature convergence, topology, biases, Real valued and binary PSO, Ant colony optimization (ACO)-Formulations, Pseudo-code. Applications of PSO and ACO. Artificial Bee Colony(ABC).

08 Hours

Text Books:

1. Principles of soft computing by S.N. Sivanandam and S.N.Deepa, 3rd Wiley India. 2019
2. Neural Networks, Fuzzy Logic and Genetic Algorithm: Synthesis and Applications" S. Rajsekaran and G.A. Vijayalakshmi Pai, 1 st Prentice Hall of India 2003

Reference books:

1. Fuzzy Logic with engineering applications Timothy J. Ross, 3rd edition, Wiley India. 2011
2. Soft Computing & Intelligent Systems Theory & Applications N. K. Sinha and M. M. Gupta 1st edition, Academic Press /Elsevier 2009

COURSE OUTCOMES:

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

CO1	Describe basic concepts of soft computing
CO2	Apply the concepts of soft computing techniques to solve engineering problems.
CO3	Analyse the evolutionary algorithms and fuzzy logic reasoning to handle uncertainty.
CO4	Demonstrate algorithms to solve combinatorial optimization problems.
CO5	

CO-PO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C13.1	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.2	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.3	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.4	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-
C13.5	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-

FUNDAMENTALS OPERATING SYSTEMS					
Course Code	**L:T:P:S**	**Credits**	**Exam Marks**	**Exam Duration**	**Course Type**
22CST643	3:0:0:0	3	CIE:50 SEE:50	3 Hours	OEC

PREREQUISITES:

Good knowledge of C, Computer Organization and Architecture, x86 Assembly level programming.

Course Objectives :

The Student will be able to:

- To Demonstrate the need for OS and different types of OS
- To discuss suitable techniques for management of different resources
- To demonstrate different APIs/Commands related to processor, memory, storage and file system management.

Teaching-Learning Process (General Instructions)

Teachers can use the following strategies to accelerate the attainment of the various course outcomes.

13. Lecturer methods (L) need not to be only traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
14. Use of Video/Animation to explain functioning of various concepts.
15. Encourage collaborative (Group Learning) Learning in the class.
16. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information rather than simply recall it.
17. Role-play for process scheduling.
18. Demonstrate the installation of any one Linux OS on VMware/Virtual Box

Syllabus

Module I

Introduction to operating systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments.

Operating System Services: User - Operating System interface; Operating System generation; System boot.

Textbook 1: Chapter – 1 (1.1-1.12), 2 (2.1,2.2,2.10,2.11) **8 Hours**

Module II

Process Management: Process concept; Process scheduling; Operations on processes; Inter process communication.

Multi-threaded Programming: Overview; Multithreading models; Thread Libraries.

Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Thread scheduling; Multiple-processor scheduling.

Textbook 1: Chapter – 4 (4.1-4.3), 5 (5.1 -5.5) **8 Hours**

Module III

Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

Textbook 1: Chapter –7 (7.1 -7.7) **8 Hours**

Module IV

Memory Management: Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.

Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing.

Textbook 1: Chapter -8 (8.1-8.6), 9 (9.1-9.6) **8 Hours**

Module V

File System, Implementation of File System: File system: File concept; Access methods; Directory and Disk structure; File system mounting; File sharing; **Implementing File system:** File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.

Textbook 1: Chapter – 10 (10.1-10.5) ,11 (11.1-11.5), **8 Hours**

Course outcomes:
The Student will be able to:

CO 1. Explain the structure and functionalities of operating system.

CO 2. Apply appropriate algorithms for the given problem.

CO 3. Analyze the various techniques used in operating system design and implementation.

CO 4. Compare various types of operating system based on different attributes.

CO 5. Prepare a comparative report on functionalities of operating system.

Text Books:

3. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 8th edition, Wiley-India, 2015

Reference Books:

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

E-Resources:

9. https://youtu.be/mXw9ruZaxzQ
10. https://youtu.be/vBURTt97EkA
11. https://www.tutorialspoint.com/operating_system/index.htm.
12. https://www.studytonight.com/operating-system/.

CO-PO-PSO Mapping

PO'S	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO1 0	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO2	3	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO3	-	3	-	-	-	-	-	-	-	-	-	-	3	-	-
CO4	-	-	2	-	-	-	-	-	-	-	-	-	3	-	-
CO5	-	-	-	-	-	-	-	-	-	2	-	2	3	-	-
CO	3	3	2	-	-	-	-	-	-	2	-	2	3	-	-

INTRODUCTION TO WEB PROGRAMMING

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
22CST644	3 : 0 : 0: 0	3	CIE:50 SEE:50	3 Hours	OEC

Description of the course: This course provides an introduction to web development and client-side scripting. After providing a review of HTML5 and CSS, the course provides exposé to the concepts of web programming using client side scripting. The course covers basic construction of web page, cascading style sheet, and java script.

Prerequisite:

Basic programming and debugging skills, Java Programming, Database and SQL queries, connectivity of front end and back end

Course Objectives: This course will enable students to:

- Recollect the evolution of World Wide Web and its relevance to today's technological revolution and also, to comprehend HTML, CSS scripts to design web layouts.
- Aquire Java Script skills for developing client side web applications.
- Interpret the use of jQuery libraries to simplify complicated JavaScript applications and also, to perform DOM manipulation using jQuery constructs.
- Cognize Bootstrap framework with a focus on creating interactive and responsive web pages.
- Assimilate XML fundamentals for developing applications over web.

Syllabus

Module - I

HTML 5 and CSS: Introduction to Hyper Text Markup Language, HTML Elements and Attributes, Headers, Colors, Formatting Elements, Links, Images, Tables, Divs, Lists, Forms, Frames, iframes, HTML Media. **CSS**: Introduction to CSS, CSS selector, CSS formatting, positioning, layouts, debugging. **08 Hours**

Module - II

JavaScript: Introduction, Scripts and HTML Document, JS Output Statements, Variables, Data Types and Conversions, Operators, Expressions, Control Structure, Decisions and Loops, Functions, Document Object Model, Forms and Form Handling Elements, Scripting, Event Handling, Regular Expressions, WEB SQL database

08 Hours

Module-III

JQuery: Introduction, Selectors, Events, jQuery DOM Manipulation: jQuery HTML, jQuery CSS, jQuery Event Model, jQuery Effects and Animations, jQuery Plugins.

08 Hours

Module - IV

Bootstrap: Bootstrap Scaffolding, Bootstrap CSS, Bootstrap Layout Components, Bootstrap JavaScript Plugins, Using Bootstrap, Web services **08 Hours**

Module – V

XML: Introduction, Syntax, Document Type Definitions, Namespaces, XML Schemas, Displaying Raw XML Documents, Displaying XML Documents with CSS. **08 Hours**

Course Outcomes

Sl. No.	Course Outcomes
CO1	Analyze the web layouts with style sheets and web screens in a presentable form.
CO2	Create interactive web pages through form validations.
CO3	Design scripts using JavaScript in a web page. .
CO4	Integrate responsive webpages for frameworks.
CO5	Develop applications by using synchronous and asynchronous communication over web.

Text Books:

1. Robert W. Sebesta: "Programming the World Wide Web", Pearson, 4th Edition, 2012, ISBN: 978-81-317-6458-9.

2. Jon Duckett: "Web Design with HTML, CSS, JavaScript and jQuery Set", Wiley, 1st Edition, 2014, ISBN 13: 978-1118907443.

3. Silvio Moreto, Matt Lambert, Benjamin Jakobus, Jason Marah:" Bootstrap 4 - Responsive Web Design", Packt Publishing, 2016, ISBN 978-1-78839-731-5

Reference Books:

1. Jake Spurlock: "Bootstrap, Shroff", O"Reilly Media, United States of America, 1st Edition, 2013, ISBN: 978 -1 -4493-4391-0.

2. Bear Bibeault, Yehuda Katz and Aurelio De Rosa: "jQuery in Action", Dreamtech Press, New Delhi, India, 3rd Edition, 2015, ISBN: 978-1617292071

E-Resources:

1. http://www.w3schools.com/
2. https://www.tutorialspoint.com/
3. https://www.javascript.com/learn/
4. https://learn.jquery.com

CO-PO & PSO Mapping:

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO's		
													PSO1	PSO2	PSO3
C212.1	-	-	2	-	-	-	-	-	-	-	-	-	2	-	-
C212.2	-	-	2	-	-	-	-	-	-	-	-	-	2	-	-

C212.3	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**-**	**-**
C212.4	**-**	**-**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**-**	**-**
C212.5	**-**	**-**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**-**	**-**
C212	**3**	**-**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**-**	**-**

Course Name: DevOps

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
22CS673	1:0:0:0	01	CIE:50 SEE:50	03 Hours	PEC

Course Objectives:

This course will enable students to:

- Understand and Apply DevOps Principles and Practices.
- Analyse the Impact of Cloud Platforms on DevOps Practices.
- Design and Implement Effective Architectural Structures for DevOps.
- Develop and Manage Deployment Strategies and Monitoring Practices.
- Implement Security Measures and Continuous Deployment Pipelines for Enterprises.

Syllabus

MODULE- I

What Is DevOps?: Introduction, Why DevOps?, DevOps Perspective, DevOps and Agile, Team Structure, Coordination, Barriers

The Cloud as a Platform: Introduction, Features of the Cloud, DevOps Consequences of the Unique Cloud Features.

08 Hours

MODULE - II

Overall Architecture: Do DevOps Practices Require Architectural Change?, Overall Architecture Structure, Quality Discussion of Microservice Architecture, Microservice Adoption for Existing Systems,

Building and Testing: Introduction, Moving a System Through the Deployment Pipeline, Crosscutting Aspects, Development and Pre-commit Testing, Build and Integration Testing, UAT/Staging/Performance Testing, Production

08 Hours

MODULE - III

Deployment: Introduction, Strategies for Managing a Deployment, Logical Consistency, Packaging, Deploying to Multiple Environments, Partial Deployment, Rollback, Tools.

Monitoring: Introduction, What to Monitor, How to Monitor, When to Change the Monitoring Configuration, Interpreting Monitoring Data, Challenges, Tools, Diagnosing an Anomaly from Monitoring Data—the Case of Platformer.com.

08 Hours

MODULE- IV

Security and Security Audits: What Is Security?, Threats, Resources to Be Protected, Security Roles and Activities, Identity Management, Access Control, Detection, Auditing, and Denial of Service, Development, Auditors, Application Design Considerations, Deployment Pipeline Design Considerations.

08 Hours

MODULE- V:

Implementing a Continuous Deployment Pipeline for Enterprises: Introduction, Organizational Context, The Continuous Deployment Pipeline, Baking Security into the Foundations of the CD Pipeline, Advanced Concepts.

Migrating to Microservices: Introduction to Atlassian, Building a Platform for Deploying Microservices, BlobStore: A Microservice Example, Development Process, Evolving BlobStore

08 Hours

Text Books:

3. DevOps: A Software Architect's Perspective by Len Bas, Ingo Weber, Liming Zhu, Addison-Wesley Pearson, 2015

Reference books:

3. Learning DevOps by Mikael Krief, Second Edition, Packt Publishing, 2022

COURSE OUTCOMES:

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

CO1	Develop and Implement Effective DevOps Strategies and Practices
CO2	Analyze Cloud Platform Features and Their Implications for DevOps
CO3	Design and Evaluate Architectural Models for DevOps Implementation
CO4	Develop and Manage Deployment Pipelines and Monitoring Solutions
CO5	Implement Security Measures and Continuous Deployment Pipelines for Enterprises

CO-PO MAPPING:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3
C13.1	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.2	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.3	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.4	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-
C13.5	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-

VII SEMESTER

COURSE SYLLABUS

Course Code	L : T : P : S	Credits	Exam Marks	Exam Duration	Course Type
22CST73	3 : 0 : 0 : 0	3	CIE:50 SEE:50	3 Hours	Theory

Course Objectives:

As a student will be able to learn:

- Acquire the knowledge of basic concepts of cryptography and network security and classify attacks on a network.
- Understand and analyze the different processes for hiding the information with conventional cryptographic algorithms.
- Comprehend various block cipher cryptosystems.
- Learn the concepts of public cryptosystems and key management Systems.
- Understand and apply authentication techniques to provide secure communication.

Prerequisites:

Students should have the knowledge of Computer Networks, Mathematics, and Algorithm Concepts.

Syllabus

Module – I

Introduction: Service mechanisms and attacks, The OSI security architecture, A Model for Network Security. Symmetric cipher model, substitution techniques- Transposition techniques, Steganography.

08 Hours

Module – II

Block Ciphers and DES: Traditional Block cipher structure, Simplified DES. Block cipher principles, DES, Strength of DES, Block cipher design principles.

08 Hours

Module – III

Advanced Encryption Standard - AES Structure, Transformation Function: Substitute Bytes Transformation, Shift Row Transformation, Mix Column Transformation, Add Round Key Transformation, AES key expansion.

08 Hours

Module – IV

Asymmetric Ciphers - Public key cryptography and RSA: Principles of public key cryptosystems, RSA algorithm. **Other public key cryptosystems and key management:**
, Diffie-Hellman key exchange. Elliptic Curve Cryptography.

08 Hours

Module – V

Network Security Applications - Authentication Applications: X.509 Authentication Service, Kerberos.

08 Hours

Course Outcomes:

At the end of this course, students will be able to

CO1: Describe the basic concepts of cryptography and network security and classify attacks on a network, symmetric ciphers and substitution techniques.

CO2: Apply and integrate the different process for hiding the information with conventional cryptographic algorithms, transposition techniques and block ciphers.
CO3: Illustrate the various block cipher cryptosystems like DES and AES.
CO4: Analyse public cryptosystems and key management systems
CO5: Analyse and demonstrate authentication techniques to provide secure communication.

Textbooks:

1) Cryptography and Network Security: William Stallings, Pearson Education, 2003
2) Behrouz A Forouzan, Debdeep Mukhopadhyay: Cryptography and Network Security, 2nd Edition, Special Indian edition, Tata McGraw-Hill, 2011.

Reference Books:

1) Cryptography and Network Security, Atul Kahate, TMH, 2003

Reference Online Resources:

1) https://nptel.ac.in/course.php

Course Name: Deep Learning

Course Code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Type
22CST744	3:0:0:0	03	CIE:50 SEE:50	03 Hours	PEC

Course Objectives:

This course will enable students to:

- Understand the theoretical foundations of deep learning.
- Explore various neural network architectures.
- Evaluate deep learning models for different tasks.
- Design and implement deep learning models.
- Develop projects using deep learning frameworks such as TensorFlow and PyTorch.

Syllabus

MODULE- I

Introduction to Deep Forward Networks: Learning XOR, Gradient-Based Learning, Hidden Units, Architecture Design, Back-Propagation and Other Differentiation Algorithms.

08 Hours

MODULE - II

Regularization for Deep Learning: Parameter Norm Penalties, Norm Penalties as Constrained Optimization, Regularization and Under-Constrained Problems, Dataset Augmentation, Semi-Supervised Learning, Multi-Task Learning, Early Stopping, Parameter Tying and Parameter Sharing, Sparse Representations, Bagging and Other Ensemble Methods, Dropout, Adversarial Training,

08 Hours

MODULE - III

Convolutional Networks: The Convolution Operation, Motivation, Pooling, Convolution and Pooling as an Infinitely Strong Prior, Variants of the Basic Convolution Function, Structured Outputs, Data Types, Efficient Convolution Algorithms, Random or Unsupervised Features, The Neuroscientific Basis for Convolutional Networks, Convolutional Networks and the History of Deep Learning

08 Hours

MODULE- IV

Sequence Modeling: Recurrent and Recursive Nets: Unfolding Computational Graphs, Recurrent Neural Networks, Bidirectional RNNs, Encoder-Decoder Sequence-to-Sequence Architectures, Deep Recurrent Networks, Recursive Neural Networks, The Challenge of Long-Term Dependencies, Echo State Networks, Leaky Units and Other Strategies for Multiple Time Scales, The Long Short-Term Memory and Other Gated RNNs, Explicit Memory,

08 Hours

MODULE- V:

Autoencoders: Undercomplete Autoencoders, Regularized Autoencoders, Representational Power, Layer Size and Depth, Stochastic Encoders and Decoders, Denoising Autoencoders, Learning Manifolds with Autoencoders, Contractive Autoencoders, Predictive Sparse Decomposition, Applications of Autoencoders

, 08 Hours

Text Books:

4. Ian Goodfellow, Yoshua Bengio, Aaron Courville - Deep Learning (2017, MIT),

Reference books:

4. Deep Learning with Python by François Chollet, Second Edition, Manning Publication 2021
5. Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow, by Aurélien Géron, 2nd edition, O'Reilly Publications, 2019

COURSE OUTCOMES:

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

CO1	Understand the theoretical foundations of deep learning
CO2	Explore various neural network architectures.
CO3	Evaluate deep learning models for different tasks
CO4	Design and implement deep learning models for various tasks.
CO5	Develop projects using deep learning frameworks such as TensorFlow and PyTorch.

CO-PO MAPPING:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3
C13.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C13.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

SOFTWARE ENGINEERING

Course Code	L:T:P	Credits	Exam Marks	Exam Duration	Course Type
22CST753	3:0:0	3	CIE:50 SEE:50	3 Hours	OEC

PREREQUISITES:

Basic understanding of Object Oriented Concepts.

Course Objectives :

The Student will be able to:

- Understand the basic principles and practices of Software Engineering.
- Gain Knowledge of Software Engineering Design Techniques and practices for developing software.
- Learn the various requirements, design and Testing Techniques to select the appropriate techniques for the software system.
- Learn Models for different phases of software development to solve real world problems.
- Gain knowledge of Projects by Estimating cost and time required for developing the Software Product.

Syllabus

Module I

Introduction:
Professional software development, Software engineering ethics, Case studies.
Software Processes: Software Process models: The Waterfall model – A Case study, Incremental development, Reuse-oriented software engineering, Process activities: Software specification, Software design and implementation, Software validation, Coping with Change: Prototyping, Incremental Delivery, Boehm's Spiral Model.

Textbook 1: Chapter 1: 1.1 to 1.3
Textbook 1: Chapter 2: 2.1 to 2.3 **08 Hours**

Module II

Requirements Engineering: Functional and non-functional requirements: Functional requirements. Non-functional requirements, Introduction to Requirements specification.
Agile Software Development: Agile methods- Plan driven and Agile Development, Introduction to Extreme Programming.

Textbook 1: Chapter 3: 3.1 to 3.3
Textbook 1: Chapter 4: 4.1 to 4.4 **08 Hours**

Module III

Design and Implementation: Object-oriented design using UML: System Context and Interaction, Architectural design, Object Class identification, design Models, Interface Specification, Design Patterns, Implementation issues, Open Source development.

Textbook 1: Chapter 7: 7.1 to 7.4 **08 Hours**

Module IV

Software Testing: Development Testing: Unit Testing, Choosing Unit Test Cases, Component Testing, System Testing, Test Driven Development, Release Testing: Requirements Based Testing, Scenario Testing, Performance Testing, User Testing. A Demo of Selenium**.**

Textbook 1: Chapter 8: 8.1 to 8.4 **08 Hours**

Module V

Introduction to Project Management:

Introduction, Project and Importance of Project Management, Contract Management, Activities Covered by Software Project Management, Plans, Methods and Methodologies, Some ways of categorizing Software Projects, Stakeholders, Setting Objectives, Business Case, Project Success and Failure, Management and Management Control, Project Management life cycle.

Textbook2:Chapter1:1.1 to 1.17 **08 Hours**

Course outcomes:

The Student will be able to:

CO1: Describe the basic principles and practices of Software Engineering.

CO2: Apply Software Engineering Design Techniques and practices for developing a Software.

CO3: Analyze the various requirements, design and Testing Techniques to select the appropriate techniques for the software system.

CO4: Design Models for different phases of software development to solve real world problems.

CO5: Manage Projects by Estimating cost and time required for developing the Software Product.

Text Books:

1. Ian Summerville: Software Engineering, 9th Edition, Pearson Education, 2017. **Chapter 1: 1.1 to 1.3, Chapter 2: 2.1 to 2.3**, **Chapter 3: 3.1 to 3.3, Chapter 4: 4.1 to 4.4, Chapter 7: 7.1 to 7.4 and Chapter 8: 8.1 to 8.4.**
2. Bob Hughes, Mike Cotterell, Rajib Mall: Software Project Management, 6th Edition, McGraw Hill Education, 2017. **Chapter1:1.1 to 1.17.**

Reference Books:

1. Roger S. Pressman: Software Engineering-A Practitioners approach, 9th Edition, TataMcGraw Hill, 2023.
2. Pankaj Jalote: An Integrated Approach to Software Engineering, 3rd Edition, Wiley India, 2005.
3. Deepak Gaikwad, Viral Thakkar, DevOps Tools from Practitioner's Viewpoint, 3rd Edition, Wiley India, 2020.
4. Michael Blaha, James Rumbaugh: Object Oriented Modelling and Design with UML, 2nd Edition, Pearson Education,2005.

Assessment Details (both CIE and SEE)

Evaluation Type	Component	Max. Marks	Marks Reduced To	Min Marks	Evaluation Details
Internal Assessment Test (IAT)	IAT 1	25	**25**	**20**	Average of two IATs, Scaled down to 25 Marks
	IAT 2	25			
Comprehensive Continuous Evaluation (CCE)	CCE-1	25	**25**		Minimum of two Assessment Methods as per 22OB4.2 of regulation. Average of CCEs, Scaled down to 25
	CCE-2	25			
Total CIE		**-**	**50**	**20**	Scaled-down Marks of IAT and CCE to 25
SEE		**100**	**50**	**18**	Conducted for 100 Marks and Scaled down to 50
CIE + SEE		**-**	**100**	**40**	

E-Resources:

1. https://onlinecourses.nptel.ac.in/noc20_cs68/preview
2. https://www.youtube.com/watch?v=WxkP5KR_Emk&list=PLrjkTql3jnm9b5nr-ggx7Pt1G4UAHeFlJ
3. http://elearning.vtu.ac.in/econtent/CSE.php
4. http://elearning.vtu.ac.in/econtent/courses/video/CSE/15CS42.html
5. https://nptel.ac.in/courses/128/106/128106012/
6. NPTEL: https://nptel.ac.in/courses/106105182
7. SWAYAM: https://onlinecourses.swayam2.ac.in/cec20_cs07/preview
8. IIT Chennai: https://onlinedegree.iitm.ac.in/course_pages/BSCCS3001.html

CO-PO-PSO Mapping

PO'S	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO1 0	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	-	-	-	-	-	-	-	-	-	1	-	3	-	-
CO2	3	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO3	-	3	-	-	-	-	-	-	-	-	-	-	3	-	-
CO4	-	-	3	-	-	-	-	-	-	-	-	-	3	-	-
CO5	-	-	-	1	-	-	-	-	-	-	1	1	3	-	-
CO	3	3	3	1	-	-	-	-	-	-	1	1	3	-	-