



TEERTHANKER MAHAVEER UNIVERSITY

(Established under Govt. of U. P. Act No. 30, 2008)

Delhi Road, Moradabad (U.P)

Website: www.tmu.ac.in

COURSE FILE

Faculty of Engineering

DEPARTMENT OF COMPUTER SCIENCE& ENGINEERING

Course Name	:	Software Engineering
Course Code	:	ECS404
Faculty Member	:	Namit Gupta
Program	:	B.Tech CSE
Semester	:	IV
Academic Year	:	2023-24



Index

SESSION-2024-2025

SEM- 4h

Subject: Software Engineering

Code: ECS404

Dept. /Sec: CSE

Course & Faculty Details	Page No.
Vision & Mission of the Institute	
Vision & Mission of Department	
Program Educational Objectives	
Program Outcomes	
Program Specific Outcomes	
Institute Academic Calendar	
Department Academic Calendar	
Course Evaluation Scheme	
Course Syllabus as per university	
Course Outcomes	
CO-PO Mapping	
CO-PO Mapping with Justification	
Lecture Plan & Course Coverage (E-Contents and Weblink)	
Time Table-Faculty	
Time Table- Section wise HANDOUT	
Student List (Section wise)	
List of Assignments/Quizzes	
Class Test Paper (CT1, CT2, and CT3)	
Class Test Marks (CT1, CT2, and CT3)	
Class Test Solution	
Assignments/Quizzes Solution	
Class Attendance Sheet (Month Wise)	
List of Slow Learner Students of Class test and action taken	
List of Fast Learner Students of Class test and action taken	
Previous Year Question Papers	
Question Bank	
Final Internal Marks	
End Semester Exam Analysis	
End Sem Question Paper	
Result & Analysis	
CO Attainment of End Sem	
Direct CO Attainment- CTs & End Sem	
Indirect CO Attainment- Course Exit Survey	
Overall CO Attainment (Direct and Indirect Assessment)	
Course File Closure Report	

FACULTY OF ENGINEERING

B.Tech. CSE NBA Accreditation 2025

INSTITUTE

VISION

To be recognised as institute of excellence, promoting value based education in computer science and engineering with multidisciplinary research, innovation and consultancy services to the society.

MISSION

1. To provide holistic education through outcome based learning.
2. To promote interdisciplinary learning and research for serving the needs of entire global community.
3. To inculcate innovation in graduates using technological advancements for the development of computational solutions.
4. To prepare graduates to be continuous learners by inculcating people skills.

DEPARTMENT

VISION

To be recognised as department of excellence, promoting value-based education in computer science and engineering with multidisciplinary research, innovation and consultancy services to the society.

MISSION

1. To provide value based education by implementing outcome-based teaching and learning process.
2. To promote inter-disciplinary research that for serving the needs of entire global community.
3. To inculcate desire for innovation in graduates using technological advancements for the development of effective solutions.
4. To prepare graduates as continuous learners by inculcating professional skills and ethical responsibilities.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

1. **Fundamental:** To prepare the graduates for career and proficiency in Computer Science and Engineering.
2. **Industry Ready:** To imbibe technical and aptitude skills related to Computer Science and Engineering to provide the best solution to industrial and societal problems.
3. **Continuous Learning:** To prepare graduates for pursuing higher education and life-long learning.
4. **Entrepreneurship Skills:** To imbibe entrepreneurship, leadership, effective communication and business skills through innovation, incubation and start-up activities.
5. **Social and Moral Ethics:** To develop awareness among the graduates about the various social responsibilities related to engineering ethics, human values and ecological understanding.

PROGRAM-SPECIFIC OUTCOMES (PSO)

PSO–1: Developing skills for creating computational solutions with emerging technologies, programming languages, mathematical foundations, algorithmic principles and open source platforms to solve complex engineering problems.

PSO–2: Understanding the evolutionary changes in computing by applying standard practices and skills acquired through computer engineering to provide solutions with innovative ideas and interdisciplinary research.

PSO–3: Applying modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur and pursue higher studies in their respective domains

Program Outcomes for Engineering:

PO – 1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO – 2	Problem analysis& Solving: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO – 3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO – 4	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.
PO – 5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
PO – 6	Social Interaction & effective citizenship: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO – 7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO – 8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO – 9	Attitude (Individual and team work): Function effectively as an individual, and as member or leader in diverse teams, and in multidisciplinary settings.
PO – 10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large such as, being able to comprehend and

write effective reports and design documentation, make effective presentations, and give and receive clean instructions.

PO – 11

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 projects and in multidisciplinary environments.

PO – 12

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.



TEERTHANKER MAHAVEER UNIVERSITY

(Established under Govt. of U. P. Act No. 30, 2008)

(Established under Govt. of U. P. Act No. 30, 2008)

Delhi Road, Moradabad (U.P.)

E-Mail: registrar@tmu.ac.in

E-Mail: registrar@tmu.ac.in

Phone No.: 9837933666

Ref. : TMU/R.O./2023-24/Academic Calendar/001

Date : 08.07.2023

Circular

Subject: Academic Calendar Session 2023-24 (Except Medical & Dental)

The Academic Calendar of the University for various programmes (except for Medical & Dental programmes) is given below. The dates mentioned may change depending upon the unforeseen situations that may arise owing to causes beyond control. It is advised not to deviate from the Academic Calendar session 2023-24 as given below. All the Colleges/departments (except Medical and Dental) are required to follow the Academic Calendar as below:

For Semester System

S.No.	Events/Particulars	Dates	
Odd semester			
		For 1 st Year	2 nd Year Onwards

1	Course allocation and tagging on ERP for odd semester		July 15,2023	July 01- July 15 ,2023
2	Submission of time table and upload on ERP for odd Semester		July 25,2023	
3	Upload of e-contents for Odd semester on website (if not available)		July 31,2023	
4	Commencement of classes:	Nursing Programmes	August 01,2023	August 08,2023
		All other Programmes (Including Student Induction Program for 1-3 weeks,as applicable)	August 08,2023	
5	Registration of semester course and open elective courses (if applicable) by the students	Nursing Programmes	August 01-05, 2023	August 08-12, 2023
		All other Programmes	August 08-30, 2023	
6	Sharing of Course Handouts with Students		August 30,2023	
7	CT-I*	Nursing Programmes	September 15-20, 2023	October 16-21, 2023
		All other Programmes	October 16-21, 2023	
8	Diwali Break (for students only)		November 13-18,2023	
9	CT-II*	Nursing Programmes	October 16-21, 2023	November 20-24, 2023
		All other Programmes	November 20-24, 2023	
10	CT-III*	Nursing Programmes	November 20-24, 2023	December 05-09, 2023
		All other Programmes	December 05-09, 2023	
11	End of Odd semester Classes # (For all programmes)		December 04,2023	
12	Preparatory Leave**		December 11-16,2023	
13	Conduct of Odd semester Examinations		December 18, 2023 - January 05,2024	
14	Declaration of results of Odd semester End Examinations		Within 15 days of conduct of Examinations	

* The CT of the concerned course should be conducted within the given window as per the scheduled time table. The Principals may draft the CT schedule according to the dates given by the concerned course faculty with maximum 2 CTs on a given date. Rest of the classes should remain unaffected with regard to teaching work.

** No academic activity to be scheduled during the preparatory leaves of odd semester (December 11-16,2023) strictly.

Regular clases of Nursing programmes shall continue after CT-III till December 04, 2023.

Even Semester		
16	Course allocation and tagging on ERP for even semester	December 20,2023
17	Submission of time table and upload on ERP for even semester	January 05,2024
18	Commencement of Even Semester classes	January 11,2024
19	Registration of semester course and open elective courses (if applicable) by the students	January 11-14,2024
20	Upload of e-contents for Even semester on website (if not available)	January 13,2024
21	Sharing of Course Handouts with Students	January 14,2024
22	CT-I*	February 19-24,2024
23	Holi Break (for students only)	March 25-30,2024
24	CT-II*	April 01-06,2024
25	End of Even semester Classes	May 04,2024
26	CT-III*	May 06-11,2024
27	Preparatory Leave	May 13-18,2024
28	Conduct of Even Semester Examinations	May 20 -June 08, 2024
29	Declaration of results of Even semester End Examinations	Within 15 days of conduct of examinations
30	Summer vacations/Internship (Wherever applicable)	June 10- July 31,2024
31	Commencement of next academic session	August 01,2024

*** The CT of the concerned course should be conducted within the given window as per the scheduled time table. The Principals may draft the CT schedule according to the dates given by the concerned course faculty with maximum 2 CTs on a given date. Rest of the classes should remain unaffected with regard to teaching work.**

For Annual System

S. No	Events/Particulars	Dates
1	Course allocation and tagging on ERP	July 01- July 15 ,2023
2	Submission of time table and upload on ERP	July 15,2023
3	Commencement of Classes 1 st year Nursing Students including induction programme(01 week) for newly admitted students.	August 01,2023
	Commencement of Classes 1 st year (rest of the programme) including induction programme for newly admitted students.	August 08,2023
	Commencement of Classes (2 nd Year onwards)	August 08,2023
4	Sharing of Course Handouts with Students	August 30,2023
5	Registration of semester course and open elective courses and backlogs (if any) by the students- 1 st Year Nursing Programmes	August 01-05,2023
	Registration of semester course and open elective courses and backlogs(if any) by the 1 st Year students(rest of the programmes)	August 08-30,2023
	Registration of semester course and open elective courses and backlogs (if any) by the students- (2 nd Year onwards)	August 08-12,2023
6	Upload of e-contents on website (if not available)	Up to July 31,2023
7	Diwali Break (for students only)	November 13-18,2023
8	CT-I	October 30 ,2023 - November 4 , 2023
9	Research Break/Clinical Posting/ Field projects (outside the campus)##	December 11-16,2023
10	CT-II	February 19-24,2024
11	Holi Break (for students only)	March 25-30,2024
12	End of classes	May 4,2024
13	CT-III	May 6-May 11,2024
14	Preparatory Leave Internal end term	May 13-18,2024
15	Conduct of Examinations	May 20,2024-June 08,2024
16	Declaration of Results	Within 15 days of conduct of examination

17	Summer Vacations	June 10,2024-July 31,2024
18	Commencement of next academic session	August 01,2024

Notes-

1. The CT of the concerned course should be conducted within the given window as per the scheduled time table. The Principals may draft the CT schedule according to the dates given by the concerned course faculty with maximum 2 CTs on a given date. Rest of the classes should remain unaffected with regard to teaching work.
2. Student Induction program to be planned as per the guidelines of UGC, AICTE, INC or other statutory bodies as applicable.
3. Academic Assignments including Internship/Education Tours/Training, wherever applicable, to be planned within the duration from end of last exam till the start of the next semester.
4. 15 days orientation to Internship for Physiotherapy Students to be planned during summer break.
5. Provision for supplementary examination, wherever applicable, to be made by the examination office and

submitted to the competent authority for approval to schedule the same in line with the academic calendar.

No academic activity to be scheduled during the period December 11-16, 2023.

(Aditya Sharma)
Registrar

Copy to:

- 1.P.S. to Hon'ble Chancellor
2. P.S. to Hon'ble Vice Chairman
3. P.S. to Hon'ble Vice Chancellor
4. The Controller of Examinations
5. Director-Students' Welfare
6. Director-CTLD
7. Director-HR
8. Chief Warden
9. Joint Director -Admissions
10. Dy. Director (A/c)
11. Asst. Director (Academics)
12. Asst. Director-IQAC
13. Joint Registrar (R&D)
14. Deputy Registrar (Academics)
15. Deputy Registrar (Admin)
16. Asst. Manager-IQAC
17. Guard file

- II. 1. Dean Academics
2. Principals-Above colleges

Evaluation Scheme

<u>Study & Evaluation Scheme</u>					
<u>SUMMARY</u>					
Institute Name		COLLEGE OF COMPUTING SCIENCES & INFORMATION TECHNOLOGY			
Programme		Bachelor of Technology (Computer Science & Engineering)			
Duration		Four Years full time(Eight Semesters)			
Medium		English			
Minimum Required Attendance		75%			
<u>Credits</u>					
Minimum Credits Required for Degree		186			
Assessment:					
Evaluation			Internal	External	Total
Theory			40	60	100
Practical/ Dissertations/ Project Reports/ Viva-Voce			50	50	100
Class Test-1	Class Test-2	Class Test-3	Assignment(s)	Attendance & Participation	Total
Best two out of three					
10	10	10	10	10	40
Duration of Examination			External	Internal	
			3 Hours	1.5 Hours	

To qualify the course a student is required to secure a minimum of 45% marks in aggregate including the semester end examination and teachers continuous evaluation.(i.e. both internal and external).A candidate who secures less than 45% of marks in a course shall be deemed to have failed in that course. The student should have at least 45% marks in aggregate to clear the semester.

Evaluation of practical course [From Second Year Onwards]

Practical Courses Internal Evaluation (50 marks)			
TWO EXPERIMENTS	ATTENDANCE	VIVA	TOTALINTERNAL
30	10	10	50
Practical Courses External Evaluation (50 marks)			
TWO EXPERIMENTS	FILE WORK	VIVA	TOTALINTERNAL
30	10	10	50

Class participation 10 % (regular basis) and attendance

Individual Assignment 10% (after 20 sessions)

CT1 Exam 10 % (Refer Academic Calendar)

CT2 Exam 10 % (Refer Academic Calendar)

CT3 Exam 10 % (Refer Academic Calendar)

Note: there is a consideration of best two CT's.

Course Code: ECS404	Professional Core Course-6 B.Tech (CSE) Semester IV SOFTWARE ENGINEERING	L-3 T-0 P-0 C-3
Course Outcomes:	On completion of the course, the students will be :	
CO1.	Understanding the basic concepts of software development life cycle and various process models of software development.	
CO2.	Understanding the concepts of agile software development and the basics of requirements engineering.	
CO3.	Understanding the concepts of software design	
CO4.	Understanding the concepts of coding approach, software testing and software reliability.	
CO5.	Understanding the concepts of software maintenance & its types, reverse engineering and CASE tools.	
CO6.	Applying software design techniques with the help of flow charts, ERD and DFD.	
Course Content:		
Unit-1:	Introduction: Evolution and impact of Software Engineering, Software Development Life Cycle (SDLC) Models: Waterfall Model, Prototype Model, Spiral Model, agility and Agile Process model, Extreme programming, other process models of agile development and Tools, Layered Approach. Software Requirements Analysis and Specifications Feasibility Study, Functional and Non-Functional Requirements, Requirements Gathering, Requirement Analysis and Specifications using DFD, Data Dictionaries and ER Diagrams, Requirements documentation, Characteristics and Organization of Software Requirement Specifications (SRS)	8 Hours
Unit-2:	Software-Design and Coding: Principles; Problem Partitioning; Abstraction; Top-Down and Bottom-Up design; Structured Approach; Functional vs. Object Oriented Approach; UML, Design Specifications and Verification; Cohesion;	8 Hours

	Coupling, Distributed Software Design, User Interface Design, Coding standards and Code Review Techniques	
Unit-3:	Software Testing : Software Testing Fundamentals, SDLC Testing : Unit Testing, Integration Testing, System Testing, Regression Testing, Smoke Testing, Security Test, Stress Test, Performance Test, Functional Testing or Black Box Testing: Boundary Value Analysis, Alpha Testing, Beta Testing, and Acceptance Testing, Structural Testing or White Box Testing: Basis Path Testing, DD-Paths, Cyclomatic Complexity, Data Flow Testing, Mutation.	8 Hours
Unit-4:	Test Management: Test Cycle, Test Estimation, Test Cases, Test Scenarios Testing Tools: Static, Dynamic, Characteristics of Modern Tools and Automation.	8 Hours
Unit-5:	Software Maintenance: Updates-Upgrades-Patches-Versions,Error Reporting, Customer Support, Maintenance Process, Software Reliability: Importance, Hardware Reliability and Software Reliability, Failure and Faults, Reliability Models, Software Reuse, CASE (Computer Aided Software Engineering): Scope; Architecture; CASE Support in SDLC; Documentation; Reverse Software Engineering.	8 Hours
<u>Text Books:</u>	1. Agarwal, K.K., <i>Software Engineering</i> , New Age International. .	
<u>Referen e Books:</u>	1. Sommerville, I., <i>Software Engineering</i> , Addison-Wesley. 2. Aggarwal, K.K. & Singh, Y., <i>Software Engineering</i> , New Age International Publishers.	
	3. Boris, B., <i>Software Testing Techniques</i> , Van Nostrand Reinhold. 4. Boris, B., <i>Black-Box Testing – Techniques for Functional Testing of Software and Systems</i> , John Wiley & Sons. * Latest editions of all the suggested books are recommended.	
<u>Additional Electronic Reference Material:</u>	1. https://www.youtube.com/watch?v=Ln_LP7c23WM&list=PLbRMhDVUMngf8oZR3DpKMvYhZKga90JVt&index=1 2. https://www.youtube.com/watch?v=Z6f9ckEElsU&list=PL8751DA481F0F0D17 3. https://www.youtube.com/watch?v=KMj49syT8JM&list=PLJ5C_6qdAvBHiqw9Yc7-_vyfbBG1Bmfg_	

Course Outcomes:	On completion of the course, the students will be :
CO1.	Understanding the basic concepts of software development life cycle and various process models of software development.

**CO-PO-PSO
Mapping**

CO2.	Understanding the concepts of agile software development and the basics of requirements engineering.
CO3.	Understanding the concepts of software design
CO4.	Understanding the concepts of coding approach, software testing and software reliability.
CO5.	Understanding the concepts of software maintenance & its types, reverse engineering and CASE tools.

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1	1	-	-	-	-	-	-	-	-	-	3	2	2
CO2	1	2	1	1	-	-	-	-	-	-	-	-	3	3	2
CO3	3	3	2	1	-	-	-	-	-	-	-	-	3	3	3
CO4	3	3	2	1	-	-	-	-		-	-	-	3	2	2
CO5	3	3	2	1	-	-	-	-	-	-	-	-	3	2	3

CO-PO Mapping with Justification

Department of Computer Science and Engineering, Faculty of Engineering

Proposed CO-PO Matrix

Course Name: SOFTWARE ENGINEERING

Session: 2023-24

Course Code: ECS404

The Course outcomes (COs). On completion of the course the participants will be able to:

CO1	Understanding the basic concepts of software development life cycle and various process models of software development.
CO2	Understanding the concepts of agile software development and the basics of requirements engineering.
CO3	Understanding the concepts of software design
CO4	Understanding the concepts of coding approach, software testing and software reliability.
CO5	Understanding the concepts of software maintenance & its types, reverse engineering and CASE tools.

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1									
CO2	1	2	1	1								
CO3	3	3	2	1								
CO4	3	3	2	1								
CO5	3	3	2	1								
Average	2.2	2.4	1.6	1								

Proposed CO-PSO Matrix

CO	PSO1	PSO2	PSO3
CO1	3	2	2
CO2	3	3	2

CO3	3	3	3
CO4	3	2	2
CO5	3	2	3
Average	3	2.4	2.4

PO-1	Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO-2	Problem Analysis: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
PO-3	Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO-4	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 for complex problems
PO-5	Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO-6	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 engineering practice.
PO-7	Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO-8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO-9	Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO-10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clean instructions.
PO-11	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 projects and in multidisciplinary environments.
PO-12	Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

PSO-1	Developing skills for creating computational solutions with emerging technologies, programming languages, mathematical foundations, algorithmic principles and open source platforms to solve complex engineering problems.
PSO-2	Ability to understand the evolutionary changes in computing by applying standard practices and skills acquired through computer engineering to provide solutions with innovative ideas and interdisciplinary research.

PSO-3	Ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur and pursue higher studies in their respective domains
--------------	--

Faculty Signature

Explanation of CO-PO Matrix

Course Name: SOFTWARE ENGINEERING

Course Code: ECS404

PSO No	Program Objective	Course Outcome	Relevance	Justification
PO1	Engineering Knowledge	CO1	1(Low)	Understanding SDLC and process models
		CO2	1(Low)	SDLC concepts require strong engineering and foundational knowledge of software engineering principles.
		CO3	3(Strong)	Daily stand-ups, documentation, and communication are essential
		CO4	3(Strong)	Agile is adaptive and promotes continuous learning
		CO5	3(Strong)	Agile practices involve incremental system design
PO2	Problem Analysis	CO1	1(Low)	Analysis of different models involves problem understanding and decomposition.
		CO2	2(Medium)	Understanding agile and requirements engineering
		CO3	3(Strong)	Involves identifying and analyzing customer requirements.
		CO4	3(Strong)	Designing software requires analyzing requirements and constraints.
		CO5	3(Strong)	Design principles evolve, requiring ongoing learning.
PO3	Design/Development of solutions	CO1	1(Low)	Selecting or customizing a process model involves design thinking.
		CO2	1(Low)	Central to designing architecture, components, and interfaces.
		CO3	2(Medium)	Understanding software design
		CO4	2(Medium)	Basic investigation may be needed for model evaluation.
		CO5	2(Medium)	Testing and debugging require analytical skills.
		CO2	1(Low)	Understanding coding, testing, and reliability
		CO3	1(Low)	Some exposure to modeling tools, but not tool-intensive.
		CO4	1(Low)	Understanding evolving models requires lifelong learning mindset.

PSO	Course Outcome	Relevance	Justification
PSO1	CO1	1(Low)	Understanding agile and requirements engineering
	CO2	2(Medium)	Agile development requires programming logic, planning, and computational thinking
	CO3	3(Strong)	Agile reflects evolutionary changes and standard practices in modern software development.
	CO4	3(Strong)	Teamwork, agile roles, and tools prepare students for real-world roles and professional growth.
	CO5	2(Medium)	Understanding software design
	CO4	2(Medium)	Software design builds directly on algorithmic principles and computing foundations.
	CO5	2(Medium)	Designing for maintainability and evolution connects to the ongoing changes in computing practices.
PSO3	CO1	1(Low)	Understanding coding, testing, and reliability
	CO2	1(Low)	Strong coding and testing skills are central to creating computational solutions.
	CO3	1(Low)	Understanding how testing and reliability evolve over time builds adaptability to computing changes.
	CO4	2(Medium)	High relevance to careers in industry, research, or entrepreneurial ventures.
	CO5	2(Medium)	Maintenance and reverse engineering involve keeping pace with evolution in computing and technology.

Designed By Subject Faculty:

Faculty Signature

Reviewed By:

LESSION PLAN:

Software Engineering Lesson Plan (40 Sessions)

Session Nos.	Topics	Pedagogy / Teaching Method	Book/Web References	Session Outcome (Bloom's Taxonomy)	Course Outcome (COs)
1	Introduction to Software Engineering & Evolution	Discussion with C & T	R1 (Pressman), NPTEL Videos	L2 – Understanding	CO1
2	Impact of Software Engineering	Case Study	R1, IEEE Articles	L3 – Applying	CO1
3	Software Development Life Cycle (SDLC) Overview	Lecture with Charts	R1	L2 – Understanding	CO1
4	Waterfall Model	Chalk & Talk	R1	L2 – Understanding	CO1
5	Prototype Model	Diagram-based Teaching	R1, R2	L2 – Understanding	CO1
6	Spiral Model	Comparative Analysis	R1, R2	L4 – Analyzing	CO1
7	Agile and Agile Process Models	Interactive Discussion	R1, Agile Manifesto	L3 – Applying	CO1
8	Extreme Programming (XP)	Group Activity + Video	Agile Alliance Docs	L3 – Applying	CO1
9	Other Agile Process Models & Tools	Lab + Demo	JIRA, Trello	L3 – Applying	CO1
10	Layered Approach in SE	Diagram & Whiteboard	R1	L2 – Understanding	CO1
11	Requirements Engineering Introduction	Lecture + Real-life Example	R1, R2	L2 – Understanding	CO1
12	Feasibility Study	Case Study	R1	L3 – Applying	CO1
13	Functional and Non-Functional Requirements	Discussion	R1, SRS Templates	L2 – Understanding	CO1
14	Requirements Gathering Techniques	Brainstorming + PPT	R1	L3 – Applying	CO1
15	Requirement Analysis with DFD	Lab + Tool Demo	StarUML, Lucidchart	L4 – Analyzing	CO1
16	Data Dictionaries and ER Diagrams	Chalk & Talk	R2, Database Books	L3 – Applying	CO1
17	Writing SRS Documents	Hands-on Writing	IEEE SRS Template	L3 – Applying	CO1
18	Characteristics & Organization of SRS	Discussion	R1	L2 – Understanding	CO1

19	Principles of Software Design	Chalk & Talk	R1	L2 – Understanding	CO2
20	Problem Partitioning and Abstraction	Diagram + Case Study	R1	L3 – Applying	CO2
21	Top-Down and Bottom-Up Design	Comparative Discussion	R1	L4 – Analyzing	CO2
22	Structured Approach	Lecture	R1	L2 – Understanding	CO2
23	Functional vs. Object-Oriented Design	UML Examples	R1, UML Diagrams	L4 – Analyzing	CO2
24	UML and Design Specifications	Lab using StarUML	StarUML Tool, R2	L3 – Applying	CO2
25	Cohesion and Coupling	Lecture + Examples	R1	L2 – Understanding	CO2
26	Distributed Software Design	Group Presentation	Research Papers	L4 – Analyzing	CO2
27	User Interface Design	Interface Demo	UI Principles Docs	L3 – Applying	CO2
28	Coding Standards and Code Review Techniques	Pair Programming	Clean Code Book, R2	L3 – Applying	CO2
29	Introduction to Software Testing	Lecture	R1	L2 – Understanding	CO3
30	SDLC Testing Types: Unit to Regression	Case Study	R1	L2 – Understanding	CO3
31	Functional Testing Techniques	Activity + BVA Examples	R1	L3 – Applying	CO3
32	Alpha, Beta, Acceptance Testing	Case Study	R1	L3 – Applying	CO3
33	Structural Testing Techniques	Whiteboard & Walkthrough	R1	L4 – Analyzing	CO3
34	Cyclomatic Complexity, Data Flow, Mutation	Coding Lab	R1	L4 – Analyzing	CO3
35	Test Management and Life Cycle	PPT + Chart	R1	L2 – Understanding	CO4
36	Test Estimation, Cases and Scenarios	Real-time Example	R1	L3 – Applying	CO4
37	Static vs. Dynamic Tools and Automation	Lab using Selenium	Tool Manuals, R1	L3 – Applying	CO4
38	Software Maintenance Process	Lecture	R1	L2 – Understanding	CO5
39	Reliability, Faults, Failures & Models	Case Examples	R1	L3 – Applying	CO5
40	Software Reuse, CASE Tools & Reverse Engineering	Lab + Review Session	R1, Online CASE Tools	L4 – Analyzing	CO5

Alignment/Mapping of COs & POs (Course Articulation Matrix)

(a) CO Mapping with Assessment tools

Course Outcomes	CT-1	CT-2	CT-3	Assignment	Project Report	Semester End Theory Exam
CO1	✓		✓	✓	NA	✓
CO2	✓		✓	✓	NA	✓
CO3	✓	✓	✓	✓	NA	✓
CO4		✓	✓	✓	NA	✓
CO5		✓	✓	✓	NA	✓

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1	-								
CO2	1	2	1	1								
CO3	3	3	2	1								
CO4	3	3	2	1								
CO5	3	3	2	1								

(b) Mapping /Alignment of COs with POs (Programme Articulation Matrix)

CO	PSO1	PSO2	PSO3
CO1	3	2	2
CO2	3	3	2
CO3	3	3	3
CO4	3	2	2
CO5	3	2	3

B.Tech (CSE) IV Semester

	9:30 –10:25	10:30– 11:25	11:30-12:25	12:30- 01:25	01:25- 2:25	2:25- 3:10	3:10-3:55
MON	Digital Library for Literature review	EHM403	ECS456 (A2) / ECS455 (A1)		L U N C H	ECS404	TGC407
TUE	ECS404	ECS401	ECS456 (A1) / ECS453 (A2)			ECS407	ECS405
WED	ECS407	ECS401	ECS456 (A2) / ECS453 (A1)			ECS406	TGC407
THU	ECS401	ECS406	EHM403	ECS405		ECS404	Programming Practice and Assessment
FRI	EHM403	ECS406	ECS407	ECS405		ECS456 (A1) / ECS455 (A2)	
SAT	Coding Practice / Self Laptop		Mentor Interaction	Library		Major Project Development	

S. No	Course Code	Subject Name	Room No	Short Name	Faculty Name
1	ECS401	Theory of Computation	3106	RG	Mr. Rupal Gupta
2	ECS407	Java Programming	3106	GKR	Mr. Gaurav Kumar Rajput
3	ECS404	Software Engineering	3106	NG	Dr. Namit Gupta
4	ECS405	Computer Based Numerical & Statistical Techniques	3106	AKS	Dr. Ashendra Kumar Saxena
5	ECS406	Operating System	3106	AJ	Mr. Aaditya Jain
6	EHM403	Management Concepts & Organizational Behavior	3106	MHS	Mohd. Saleem
7	ECS456	Java Programming (Lab)	3201A	GKR	Mr. Gaurav Kumar Rajput

8	ECS453	Computer Based Numerical & Statistical Techniques (Lab)	3201B	AKS	Dr. Ashendra Kumar Saxena
9	ECS455	OS Lab with Software Engineering (Lab)	3201B	AJ	Mr. Aaditya Jain
10	TGC407	Analytical Reasoning	3106	ANB	Mr. Anant Bharadwaj

Student List B.Tech.(CSE) Regular (2022-2026)

Sl.	Enrollment No.	Student Name
1	TCA2209001	AADI JAIN
2	TCA2209002	AADITYA JAIN
3	TCA2209006	ABHAY KUMAR
4	TCA2209007	ABHIJEET SHARMA
5	TCA2209009	ADEESH JAIN
6	TCA2209010	ADITI JAIN
7	TCA2209011	ADITYA ANISH JAIN
8	TCA2209014	ADITYA WAGHE
9	TCA2209015	AGAM JAIN
10	TCA2209018	AKHIL CHAUHAN
11	TCA2209020	AMAN CHAUHAN
12	TCA2209023	AMAN KUMAR
13	TCA2209028	ANEKANT JAIN
14	TCA2209032	ANSH KUMAR VASHISHTHA
15	TCA2209036	ARJIT PHOGAT
16	TCA2209039	ARPITA SRIVASTAVA
17	TCA2209041	ARYAN JAIN
18	TCA2209047	BHANU PRATAP SINGH
19	TCA2209048	BHARTI SHARMA
20	TCA2209051	BHUMIKA JAIN
21	TCA2209052	DAKSH JAIN
22	TCA2209058	DEEYA JAIN
23	TCA2209059	DEVANSH BHARADWAJ
24	TCA2209062	DIVYANSH JAIN
25	TCA2209063	DRASHTI JAIN
26	TCA2209065	HAMZA FARHAT
27	TCA2209069	HARSHIT KUMAR
28	TCA2209070	HEMANT JAIN
29	TCA2209071	HEMANT PANDEY
30	TCA2209075	JAIN AMAN AJAY
31	TCA2209076	JANVI CHAUHAN
32	TCA2209079	KASHISH JAIN
33	TCA2209080	KASHISH RASTOGI
34	TCA2209085	LAKI SAINI
35	TCA2209088	MANSI CHOUDHARY
36	TCA2209107	NITISHA JAIN
37	TCA2209109	NAMAN JAIN
38	TCA2209112	NIKHIL JAIN

39	TCA2209115	NISHANT CHAUHAN
40	TCA2209120	PAYAL JAIN
41	TCA2209122	PRAKASH RAJ
42	TCA2209126	PRASHANT PATEL
43	TCA2209133	RAHUL KUMAR
44	TCA2209140	RAVIKANT SHARMA
45	TCA2209146	RITIK KUMAR
46	TCA2209147	RIYA JAIN
47	TCA2209149	ROHAN KANCHAN SAXENA
48	TCA2209155	SAMARTH SAXENA
49	TCA2209173	SHREYANSH JAIN
50	TCA2209175	SUBRAT JAIN
51	TCA2209180	TANISHKA JAIN
52	TCA2209181	TANVI JAIN
53	TCA2209184	UDAY JAIN
54	TCA2209185	UDIT JAIN
55	TCA2209187	VANSHIKA TRIPATHI
56	TCA2209193	YAMINI JAIN
57	TCA2209194	YUVRAJ SINGH
58	TCA2209204	LAXMI
59	TCA2209094	MOHD GUL SHEEMAB
60	TCA2209102	MOHD SANIB

**Student List B.Tech.(CSE) Lateral
(2023-2026)**

S.No.	Enrollment No.	Student Name
1	TCA2311002	ANKUR JAIN
2	TCA2311003	AYAZ ARIF KHAN
3	TCA2311004	AYUSHI BHATNAGAR
4	TCA2311005	BHANU PRATAP
5	TCA2311007	RAJ PRATAP
6	TCA2311008	YOGESH VERMA

Year: 2nd	Semester: 4	Academic Session: 2023-24
Course Code: ECS404	Course Name: SOFTWARE ENGINEERING	
		Max. Marks: 20

*Bloom Level {1-Remembering, 2-Understanding, 3-Applying, 4-Analyzing, 5-Evaluating, 6-Creating}

Faculty of Engineering

Assignment -1

Program Name: B. Tech. (CSE)

Attempt all questions.

		MM	Unit	BTL	CO
1	Explain the differences between the Waterfall Model and the Spiral Model in terms of their structure and risk handling.	4	1	2	1
2	Define Functional and Non-Functional Requirements with suitable examples for each in a software system.	4	1	2	2
3	What is the role of a Feasibility Study in Software Requirements Analysis? Briefly explain the types of feasibility considered.	4	1	1	2
4	Draw and explain a Level 0 Data Flow Diagram (DFD) for a Library Management System.	4	1	3	6
5	Describe any two key characteristics of a good Software Requirement Specification (SRS) document. Why are these characteristics important?	4	1	2	2

Year: 2 nd	Semester: 4	Academic Session: 2023-24
Course Code: ECS404	Course Name: SOFTWARE ENGINEERING	
		Max. Marks: 20

*Bloom Level {1-Remembering, 2-Understanding, 3-Applying, 4-Analyzing, 5-Evaluating, 6-Creating}

Faculty of Engineering

Assignment -2

Program Name: B. Tech. (CSE)

Attempt all questions.

		MM	Unit	BTL	CO
1	Differentiate between Top-Down and Bottom-Up design approaches with the help of suitable examples.	4	2	4	3
2	Explain the concepts of Cohesion and Coupling in software design. Why is high cohesion and low coupling desirable?	4	2	2	3
3	What is the role of UML in software design? Briefly describe any two commonly used UML diagrams.	4	2	1	3
4	Compare Functional and Object-Oriented design approaches. Highlight one advantage of each.	4	2	3	3
5	What are coding standards and how do they contribute to software quality? Mention any two common code review techniques.	4	2	2	4

Year: 2nd	Semester: 4	Academic Session: 2023-24
Course Code: ECS404	Course Name: SOFTWARE ENGINEERING	
		Max. Marks: 20

*Bloom Level {1-Remembering, 2-Understanding, 3-Applying, 4-Analyzing, 5-Evaluating, 6-Creating}

Faculty of Engineering

Assignment -3

Program Name: B. Tech. (CSE)

Attempt all questions.

		MM	Unit	BTL	CO
1	Differentiate between Black Box Testing and White Box Testing with suitable examples.	4	3	2	4
2	Explain Boundary Value Analysis with an example. Why is it important in functional testing?	4	3	2	4
3	Define Alpha Testing and Beta Testing. How do they differ in terms of testing environment and user involvement?	4	3	1	4
4	What is Cyclomatic Complexity? How is it useful in White Box Testing?	4	3	3	4
5	Briefly describe any four types of system-level testing (e.g., Regression, Stress, Performance, Security).	4	3	2	4

Year: 2nd	Semester: 4		Academic Session: 2023-24
Course Code: ECS404	Course Name: SOFTWARE ENGINEERING		
		Max. Marks: 20	

*Bloom Level {1-Remembering, 2-Understanding, 3-Applying, 4-Analyzing, 5-Evaluating, 6-Creating}

Faculty of Engineering

Assignment -4

Program Name: B. Tech. (CSE)

Attempt all questions.

		M M	Uni t	BT L	CO
1	What are the key phases of a typical test cycle? Briefly describe each phase.	4	4	1	4
2	Differentiate between a Test Case and a Test Scenario with suitable examples.	4	4	3	4
3	Explain any two common methods used for test estimation in software testing.	4	4	2	4
4	Compare Static Testing and Dynamic Testing with one advantage of each.	4	4	3	4
5	List and explain any two characteristics of modern automated testing tools. Why is automation important in large projects?	4	4	2	4

Year: 2nd	Semester: 4		Academic Session: 2023-24
Course Code: ECS404	Course Name: SOFTWARE ENGINEERING		
			Max. Marks: 20

*Bloom Level {1-Remembering, 2-Understanding, 3-Applying, 4-Analyzing, 5-Evaluating, 6-Creating}

Faculty of Engineering

Assignment -5

Program Name: B. Tech. (CSE)

Attempt all questions.

		M M	Uni t	BT L	CO
1	Explain the differences between updates, upgrades, patches, and versions with suitable examples.	4	5	2	5
2	What is the role of error reporting and customer support in the software maintenance process? How do they help improve software quality?	4	5	1	5
3	Define software reliability. How does it differ from hardware reliability?	4	5	2	4
4	What is reverse software engineering? Mention its purpose and any one use- case.	4	5	2	5
5	What is the scope of CASE tools in software development? Briefly describe how CASE supports documentation and SDLC activities.	4	5	3	5

College of Computing Sciences and Information Technology

FIRST INTERNAL EXAMINATION (CT-1) [Even Semester]

Program Name: BTECH (C.S)

Year: II	Semester: 4	Academic Session: 2023-2024
Course Code: ECS404	Course Name: Software Engineering	
Duration: 90 minutes	Max. Marks: 30	

*Bloom Level {1-Remembering, 2-Understanding, 3-Applying, 4-Analyzing, 5-Evaluating, 6-Creating}

Attempt all questions.

1.	SHORT QUESTION		MM	Unit	BT L	CO
	<u>Answer any five questions</u>					
	A.	What is software engineering?	(2)	1	1	1
	B.	Differentiate between functional and non-functional requirements.	(2)	1	2	1
	C.	List any two advantages of the Waterfall Model.	(2)	1	1	1
	D.	Why is requirement analysis important in software development?	(2)	1	2	1
	E.	Differentiate between cohesion and coupling.	(2)	2	2	2
	F.	List two benefits of using UML in design.	(2)	2	2	2
	G.	Define user interface design.	(2)	2	1	2
	H.	What is the purpose of code review?	(2)	2	2	2
	LONG QUESTIONS					
2.	A	Compare and contrast the Prototype Model and Spiral Model. When would we use one over the other?	(10)	1	4	1
	OR					
	B	Draw and explain a Level-0 DFD for an online shopping system.	(10)	1	3	1
3.	A	Apply the top-down design approach to develop a software module for a library book management system.	(10)	2	3	2
	OR					
	B	Compare the object-oriented approach with the functional approach in terms of design flexibility and reusability.	(10)	2	4	2

1.	SHORT QUESTION		Max Marks	Unit	Bloom Level*	CO
	<u>Answer any five questions</u>					
	A	What is software testing?	(2)	3	1	3
	B	Differentiate between alpha testing and beta testing.	(2)	3	2	3
	C	Define boundary value analysis in black box testing.	(2)	3	2	3
	D	What is the role of integration testing in SDLC?	(2)	3	2	3
	E	Define test estimation.	(2)	4	1	4
	F	What is the difference between a test case and a test scenario?	(2)	4	2	4
	G	What are dynamic testing tools? Give an example.	(2)	4	1	4
	H	Name two tools used for automated testing.	(2)	4	1	4
	LONG QUESTIONS					
2.	A	Design and describe a unit test case for a function that calculates student grades based on percentage marks.	(10)	3	3	3
	OR					
	B	Compare stress testing and performance testing. What tools are commonly used for each, and why?	(10)	3	4	3

3.	A	Compare and contrast static testing tools and dynamic testing tools with examples and appropriate use cases.	(10)	4	4	4
OR						
	B	Discuss the pros and cons of using automation tools versus manual testing in large-scale projects.	(10)	4	2	4

College of Computing Sciences and Information Technology

THIRD INTERNAL EXAMINATION (CT-3) [Even Semester]

Program Name: BTECH (C.S)

Year: II	Semester: 4	Academic Session: 2023-2024
Course Code: ECS404	Course Name: Software Engineering	
Duration: 90 minutes	Max. Marks: 30	

*Bloom Level {1-Remembering, 2-Understanding, 3-Appling, 4-Analyzing, 5-Evaluating, 6-Creating}

Attempt all questions.

1.	SHORT QUESTION	MM	Unit	BTL	CO
	<u>Answer any five questions</u>				
A.	What do you mean by a Software Requirements Specification (SRS)?	(2)	1	2	1
B.	What is the role of verification in software design?	(2)	2	2	2
C.	What is mutation testing?	(2)	3	1	3
D.	Why is test estimation critical before starting the testing process?	(2)	4	2	4
E.	What is meant by “test planning”?	(2)	4	2	4
F.	Define a test scenario with a simple example.	(2)	4	1	4

	G.	What is software maintenance?	(2)	5	1	5
	H.	Explain hardware reliability vs. software reliability.	(2)	5	2	5
	LONG QUESTIONS					
2.	A	Explain the importance of error reporting and customer support in enhancing software quality post-deployment.	(10)	5	2	5
	OR					
	B	How hardware reliability impacts software reliability? Provide examples.	(10)	5	4	5
3.	A	What different software reliability models. Which model would you recommend for a banking system, and why?	(10)	5	2	5
	OR					
	B	Explain the role and benefits of software reuse in reducing development and maintenance costs.	(10)	5	2	5

Student List B.Tech.(CSE) Regular (2022-2026)

SNO.	ENROLLMENT NO.	STUDENT'S NAME	CT1	CT2	CT3
1	TCA2209001	AADI JAIN	17	19	20
2	TCA2209002	AADITYA JAIN	AB	25	23
3	TCA2209006	ABHAY KUMAR	9	11	24
4	TCA2209007	ABHIJEET SHARMA	22	21	22
5	TCA2209009	ADEESH JAIN	28	25	AB
6	TCA2209010	ADITI JAIN	16	25	21
7	TCA2209011	ADITYA ANISH JAIN	10	11	19
8	TCA2209014	ADITYA WAGHE	AB	23	21
9	TCA2209015	AGAM JAIN	19	14	AB
10	TCA2209018	AKHIL CHAUHAN	AB	20	20
11	TCA2209020	AMAN CHAUHAN	AB	16	17
12	TCA2209023	AMAN KUMAR	14	10	21
13	TCA2209028	ANEKANT JAIN	AB	16	12
14	TCA2209032	ANSH KUMAR VASHISHTHA	AB	16	25
15	TCA2209036	ARJIT PHOGAT	AB	13	22
16	TCA2209039	ARPITA SRIVASTAVA	13	22	24
17	TCA2209041	ARYAN JAIN	AB	25	20
18	TCA2209047	BHANU PRATAP SINGH	18	17	AB
19	TCA2209048	BHARTI SHARMA	11	19	20
20	TCA2209051	BHUMIKA JAIN	AB	13	16
21	TCA2209052	DAKSH JAIN	AB	15	19
22	TCA2209058	DEEYA JAIN	AB	18	28
23	TCA2209059	DEVANSH BHARADWAJ	25	24	AB
24	TCA2209062	DIVYANSH JAIN	25	26	AB
25	TCA2209063	DRASHTI JAIN	19	23	AB
26	TCA2209065	HAMZA FARHAT	AB	20	18
27	TCA2209069	HARSHIT KUMAR	11	AB	15
28	TCA2209070	HEMANT JAIN	AB	20	19
29	TCA2209071	HEMANT PANDEY	14	21	28
30	TCA2209075	JAIN AMAN AJAY	26	23	AB
31	TCA2209076	JANVI CHAUHAN	11	22	25
32	TCA2209079	KASHISH JAIN	27	AB	20
33	TCA2209080	KASHISH RASTOGI	14	25	27
34	TCA2209085	LAKI SAINI	AB	18	14
35	TCA2209088	MANSI CHOUDHARY	12	23	23
36	TCA2209107	NITISHA JAIN	16	23	28
37	TCA2209109	NAMAN JAIN	20	25	18
38	TCA2209112	NIKHIL JAIN	16	15	AB
39	TCA2209115	NISHANT CHAUHAN	23	19	18
40	TCA2209120	PAYAL JAIN	21	27	AB

41	TCA2209122	PRAKASH RAJ	23	16	19
42	TCA2209126	PRASHANT PATEL	24	19	AB
43	TCA2209133	RAHUL KUMAR	15	18	AB
44	TCA2209140	RAVIKANT SHARMA	20	16	26
45	TCA2209146	RITIK KUMAR	AB	13	19
46	TCA2209147	RIYA JAIN	24	25	30
47	TCA2209149	ROHAN KANCHAN SAXENA	22	22	17
48	TCA2209155	SAMARTH SAXENA	12	18	25
49	TCA2209173	SHREYANSH JAIN	24	18	18
50	TCA2209175	SUBRAT JAIN	AB	8	9
51	TCA2209180	TANISHKA JAIN	19	28	AB
52	TCA2209181	TANVI JAIN	AB	23	26
53	TCA2209184	UDAY JAIN	21	27	24
54	TCA2209185	UDIT JAIN	AB	21	18
55	TCA2209187	VANSHIKA TRIPATHI	15	21	26
56	TCA2209193	YAMINI JAIN	20	24	23
57	TCA2209194	YUVRAJ SINGH	AB	25	23
58	TCA2209204	LAXMI	25	26	AB
59	TCA2209094	MOHD GUL SHEEMAB	17	22	18
60	TCA2209102	MOHD SANIB	17	6	17
61	TCA2311002	ANKUR JAIN	24	24	AB
62	TCA2311003	AYAZ ARIF KHAN	AB	AB	1
63	TCA2311004	AYUSHI BHATNAGAR	20	25	AB
64	TCA2311005	BHANU PRATAP	13	11	26
65	TCA2311007	RAJ PRATAP	AB	16	24
66	TCA2311008	YOGESH VERMA	20	21	AB

FIRST INTERNAL EXAMINATION (CT-1) [Even Semester]

Program Name: BTECH (C.S)

SOLUTION

a. What is Software Engineering?

Software Engineering is a disciplined approach to the development, operation, and maintenance of software using engineering principles and systematic methods.

b. Differentiate between Functional and Non-Functional Requirements:

Functional Requirements

Define *what* the system should do

Example: Login, data retrieval

Non-Functional Requirements

Define *how* the system performs a task

Example: Performance, security, usability

c. List any two advantages of the Waterfall Model:

1. Simple and easy to understand and use.
 2. Phases are processed and completed one at a time, making it easy to manage.
-

d. Why is requirement analysis important in software development?

It helps identify user needs, avoid misunderstandings, and ensures that the developed system meets business goals and expectations.

e. Differentiate between Cohesion and Coupling:

Cohesion

Degree to which elements of a module belong together

High cohesion is desirable

Coupling

Degree of interdependence between modules

Low coupling is desirable

f. List two benefits of using UML in design:

1. Helps visualize the system architecture clearly.
 2. Facilitates communication among stakeholders and developers.
-

g. Define user interface design:

User Interface Design is the process of designing interfaces in software with a focus on looks, style, and ease of use to enhance user interaction.

h. What is the purpose of code review?

To ensure code quality, detect bugs early, improve code maintainability, and encourage knowledge sharing among developers.

2.

a. Compare and contrast the Prototype Model and Spiral Model. When would we use one over the other?

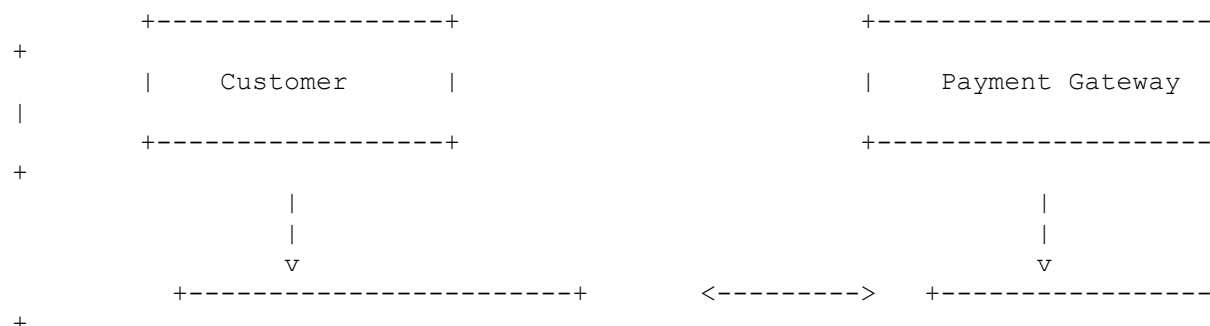
Feature	Prototype Model	Spiral Model
Focus	User feedback through working prototypes	Risk management and iterative refinement
Phases	Build quick prototype → Gather feedback → Refine	Planning → Risk Analysis → Engineering → Evaluation
Use Case	When requirements are unclear or need validation	When project is large, complex, and high-risk
Flexibility	Less structured, quick development	Highly structured with emphasis on risk control

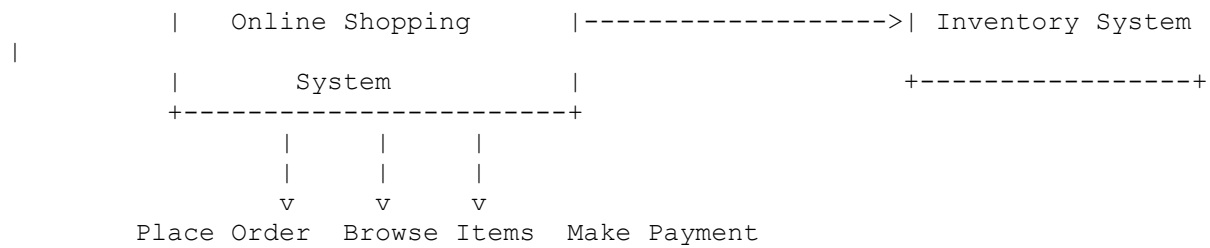
When to use:

- Use **Prototype Model** for quick requirement gathering and small projects.
- Use **Spiral Model** for large-scale projects needing risk analysis and frequent revisions.

b. Draw and explain a Level-0 DFD for an Online Shopping System

Level-0 DFD (Context Diagram)





Explanation:

- **Customer** can browse items, place orders, and make payments.
- System interacts with **Inventory System** and **Payment Gateway**.
- External entities: Customer, Inventory, and Payment Gateway.
- Single process: “Online Shopping System”.

3

a. Apply the Top-Down Design Approach to Develop a Library Book Management System

Top-Level Function:

- Library Management System

Decomposed into sub-modules:

1. **Book Management**
 - Add book
 - Remove book
 - Search book
2. **Member Management**
 - Register member
 - Remove member
 - Update member details
3. **Borrow/Return System**
 - Issue book
 - Return book
 - Calculate fine
4. **Reports**
 - Daily transaction reports
 - Overdue books

Approach: Break down the system into smaller, manageable modules starting from the highest-level functionality.

d. Compare the Object-Oriented Approach with the Functional Approach in Terms of Design Flexibility and Reusability

Criteria	Object-Oriented Approach	Functional Approach
Design Flexibility	High – supports inheritance, polymorphism, encapsulation	Moderate – depends on pure functions and recursion
Reusability	High – reusable objects and classes	Limited – functions must be explicitly reused
Modularity	Strongly modular via objects	Modular via independent functions
Maintenance	Easier due to encapsulated data and behavior	May require more effort for complex systems

SECOND INTERNAL EXAMINATION (CT-2) [Even Semester]

Program Name: BTECH (C.S)

SOLUTION

1.

a. What is software testing?

Software testing is the process of evaluating a software application to identify defects and ensure it meets the specified requirements and works as expected.

b. Differentiate between Alpha Testing and Beta Testing:

Alpha Testing	Beta Testing
Performed by internal developers/testers	Performed by real users in a real environment
Conducted before product release	Conducted after alpha testing, before final release
Controlled environment	Uncontrolled, user-end environment

c. Define boundary value analysis in black box testing.

Boundary Value Analysis (BVA) is a black box testing technique where tests are designed to include values at the boundaries of input domains (e.g., minimum, maximum, just below and above).

Example: If valid age is 18–60, test for 17, 18, 60, 61.

d. What is the role of integration testing in SDLC?

Integration testing ensures that different modules or components of a system work together correctly. It detects interface errors and interaction issues between integrated units.

e. Define test estimation.

Test estimation is the process of predicting the time, effort, and cost required to complete testing activities in a project.

f. What is the difference between a test case and a test scenario?

Test Case	Test Scenario
Detailed steps with input, execution, and expected result	High-level idea of what to test
Example: "Login with valid credentials"	Example: "Test login functionality"

g. What are dynamic testing tools? Give an example.

Dynamic testing tools evaluate software by executing code to find defects during runtime.

Example: Selenium, which automates web application testing.

h. Name two tools used for automated testing.

1. **Selenium** – for web applications
2. **JUnit** – for unit testing in Java

2.

a. Unit Test Case for a Function that Calculates Student Grades

Assume the function:

```
def calculate_grade(percentage):  
    if percentage >= 90:  
        return 'A'  
    elif percentage >= 80:  
        return 'B'  
    elif percentage >= 70:  
        return 'C'  
    elif percentage >= 60:  
        return 'D'  
    else:  
        return 'F'
```

Unit Test Case Design

Test Case ID	Input (%)	Expected Output	Description
TC01	95	'A'	Grade A for 90 and above
TC02	85	'B'	Grade B for 80–89
TC03	72	'C'	Grade C for 70–79
TC04	65	'D'	Grade D for 60–69
TC05	45	'F'	Grade F for below 60

Test Case ID	Input (%)	Expected Output	Description
TC06	90	'A'	Boundary test for Grade A
TC07	59	'F'	Boundary test for Grade F

Testing Framework Example (Python/unittest)

```
import unittest
```

```
class TestGradeCalculator(unittest.TestCase):
    def test_calculate_grade(self):
        self.assertEqual(calculate_grade(95), 'A')
        self.assertEqual(calculate_grade(85), 'B')
        self.assertEqual(calculate_grade(72), 'C')
        self.assertEqual(calculate_grade(65), 'D')
        self.assertEqual(calculate_grade(45), 'F')
        self.assertEqual(calculate_grade(90), 'A')
        self.assertEqual(calculate_grade(59), 'F')
```

```
if __name__ == '__main__':
    unittest.main()
```

b. Compare Stress Testing and Performance Testing

Feature	Stress Testing	Performance Testing
Purpose	Tests system behavior under extreme load	Tests system efficiency under expected load
Goal	Identify breaking point, system recovery	Measure responsiveness, speed, scalability
Focus	Stability, error handling under overload	Resource usage, response time, throughput
Load Condition	Beyond normal limits	Within normal or peak operating limits
Outcome	System fails gracefully or crashes	System performs within acceptable performance levels

Common Tools:

Tool	Use Case	Type
Apache JMeter	Both stress and performance	Open-source
LoadRunner	Enterprise-grade testing	Commercial
Locust	Python-based load testing	Open-source

Tool	Use Case	Type
Gatling	Scalable performance testing	Open-source

3.

a. Static Testing Tools vs Dynamic Testing Tools

Aspect	Static Testing Tools	Dynamic Testing Tools
Definition	Test software <i>without executing</i> the code	Test software <i>by executing</i> the code
Purpose	To detect errors, bugs, or code smells early	To detect run-time errors, functional bugs, and performance issues
Time of Use	Early in the development life cycle	Later stages (after code is written)
Focus	Code structure, syntax, standards	Functionality, behavior, and performance
Examples	SonarQube, ESLint, Checkstyle, Pylint	Selenium, JUnit, LoadRunner, Postman
Use Cases	Code reviews, style checks, security scans	UI testing, unit testing, integration testing, performance testing

Use Case Examples:

- **Static Tool (SonarQube):** Analyzes code quality and security issues in Java applications.
- **Dynamic Tool (Selenium):** Automates web UI testing across different browsers.

b. Automation Tools vs Manual Testing in Large-Scale Projects

Automation Testing

Pros:

- **Speed & Efficiency:** Executes tests faster and more frequently.
- **Repeatability:** Same test cases can be reused multiple times.
- **Coverage:** Supports high-volume regression and performance testing.
- **Integration:** Easily integrated into CI/CD pipelines.

Cons:

- **Initial Cost:** High setup and tool cost.
- **Skill Requirement:** Requires knowledge of scripting and tools.
- **Not Ideal for UI/UX:** Poor at exploratory or user-experience testing.

Manual Testing

Pros:

- **Flexibility:** Ideal for exploratory, usability, and ad-hoc testing.
- **Human Insight:** Detects issues automation might miss (UI glitches, user flow).
- **Low Initial Investment:** No tools or scripting needed to start.

Cons:

- **Time-Consuming:** Slower compared to automation.
- **Prone to Errors:** Manual repetition may lead to oversight.
- **Limited Coverage:** Difficult to run a large number of test cases efficiently.

When to Use What?

Scenario	Preferred Method
Frequent regression testing	Automation
New feature testing	Manual
Usability and exploratory testing	Manual
High-volume load or stress testing	Automation
Short-term project with limited budget	Manual

THIRD INTERNAL EXAMINATION (CT-3) [Even Semester]

Program Name: BTECH (C.S)

SOLUTION

1.

a. What do you mean by a Software Requirements Specification (SRS)?

A **Software Requirements Specification (SRS)** is a detailed document that describes the software system's **functional and non-functional requirements**. It serves as a foundation for design, development, and testing, ensuring that all stakeholders have a clear understanding of the system requirements.

b. What is the role of verification in software design?

Verification ensures that the software design meets the specified requirements and is free from logical errors. It helps identify problems early using techniques like reviews and inspections, answering the question:

Are we building the product right?

c. What is mutation testing?

Mutation testing is a testing technique where small, deliberate changes (mutants) are introduced into the source code to check if the existing test cases detect the errors. It evaluates the **effectiveness** of the test cases.

d. Why is test estimation critical before starting the testing process?

Test estimation is important because it helps in planning the time, resources, and budget required for testing. Accurate estimation ensures that testing is completed efficiently and that **deadlines and quality expectations** are met.

e. What is meant by “test planning”?

Test planning is the process of defining the **scope, objectives, strategy, schedule, and resources** for the testing activities. It acts as a blueprint for the testing phase and includes risk management and deliverables.

f. Define a test scenario with a simple example.

A **test scenario** is a high-level idea of what to test.

Example: Test the login functionality of an email application.

This scenario can generate multiple test cases such as:

- Valid login

- Invalid password
- Blank username, etc.

g. What is software maintenance?

Software maintenance involves updating the software after delivery to fix bugs, enhance features, adapt to new environments, or prevent issues. It includes corrective, adaptive, perfective, and preventive maintenance.

h. Explain hardware reliability vs. software reliability.

Aspect	Hardware Reliability	Software Reliability
Nature	Physical components (can wear out)	Logical code (does not physically degrade)
Failures	Due to aging, temperature, mechanical issues	Due to bugs, logic errors
Measurement	MTBF (Mean Time Between Failures)	Failure rate, reliability metrics
Fix Approach	Replace hardware	Debug and fix code

2.

a. Importance of Error Reporting and Customer Support in Enhancing Software Quality Post-Deployment

1. Error Reporting:

- **Early Bug Detection:** Allows users to report issues, helping developers identify and fix bugs quickly.
- **Real-World Feedback:** Provides insights into how the software behaves in actual usage environments.
- **Continuous Improvement:** Enables regular updates and patches, improving software stability and performance.
- **Tracking Trends:** Helps in identifying recurring issues or areas where the system is error-prone.

2. Customer Support:

- **User Guidance:** Helps users resolve issues, improving their experience and satisfaction.
- **Building Trust:** Responsive support builds user confidence in the software.
- **Prioritizing Fixes:** Support tickets help developers prioritize critical problems.
- **Feature Requests:** Users often suggest enhancements, guiding future versions of the software.

☞ Together, **error reporting and support ensure a feedback loop**, which is essential for maintaining high **software quality**, reliability, and user satisfaction.

b. How Hardware Reliability Impacts Software Reliability (with Examples)

Hardware reliability directly affects the performance and reliability of software, even if the software itself is error-free.

Impact Areas:

Hardware Issue	Impact on Software Reliability	Example
Hard Disk Failure	Loss of data, software crashes, or corrupt installations	A database application crashes due to unreadable disk blocks
Memory Errors (RAM)	Unexpected crashes, incorrect computations	A calculation module gives wrong output due to memory faults
Overheating of CPU/GPU	System slowdown, software hangs or freezes	A video editing software lags or freezes under high load
Network Interface Failures	Connectivity issues, failed data transmission	A chat app fails to send/receive messages
Power Fluctuations	Unexpected shutdowns, risk of data loss	An unsaved document is lost due to sudden power loss

3.

a. What are the different software reliability models?

Software reliability models are mathematical models used to **predict and estimate software reliability** during testing or after deployment. Some common models include:

- Jelinski-Moranda (J-M) Model**
 - Assumes a fixed number of initial faults.
 - Each failure removes one fault.
 - Failure rate decreases over time.
- Goel-Okumoto Model (Exponential Model)**
 - Assumes failures occur following a Poisson process.
 - Useful for modeling cumulative number of faults over time.
- Musa Execution Time Model**
 - Based on actual execution time rather than calendar time.
 - Suitable for real-time systems.
- Weibull Model**
 - Uses Weibull distribution to represent failure data.
 - Can model increasing, constant, or decreasing failure rates.

5. Littlewood–Verrall Model

- Probabilistic model using Bayesian methods.
- Suitable when limited failure data is available.

Which model would you recommend for a banking system, and why?

Recommended Model: Goel-Okumoto Model

Reason:

- Banking systems require **high reliability and continuous operation**.
- The **Goel-Okumoto model** is:
 - Effective in **estimating the number of software failures over time**.
 - Ideal for **mission-critical and transaction-heavy systems** like banking.
 - Supports **early failure prediction and resource planning** during testing.

It helps to plan rigorous testing and estimate when the software reaches acceptable reliability before going live.

b. Explain the role and benefits of software reuse in reducing development and maintenance costs.

Role of Software Reuse:

- **Software reuse** is the practice of using existing software components (like libraries, modules, or code frameworks) in new applications.
- It supports **modularity, standardization, and faster development**.

Benefits:

Benefit	Explanation
Reduced Development Time	Reusing pre-tested components speeds up the development process.
Lower Costs	Minimizes the need to build from scratch, saving effort and budget.
Improved Quality	Reused components are often well-tested, reducing bugs and defects.
Easier Maintenance	Standardized and modular components simplify debugging and updates.
Consistency Across Projects	Promotes uniform design and architecture across multiple projects.
Faster Time to Market	Enables quicker delivery of products and services.

7	TCA2109009	AMAN BARANWAL	P	P	P	P	P	P	P	P	P
8	TCA2109010	AMIT KUMAR	P	P	P	P	P	P		P	P
9	TCA2109011	AMITESH JHA	P	P	P	P	P	P	P	P	P
10	TCA2109014	ANSHIKA JAIN	P	P		P	P	P	P	P	P
11	TCA2109015	ARINJAY JAIN	P	P	P	P	P	P	P	P	P
12	TCA2109016	ARPIT JAIN		P	P	P	P	P	P		P
13	TCA2109017	ARUN JAIN	P	P	P	P	P	P		P	P
14	TCA2109018	ASHISH KUMAR	P	P	P	P		P	P	P	P
15	TCA2109019	ASHMIT KUMAR SAXENA	P	P		P	P	P	P	P	P
16	TCA2109020	ASTHA PANDEY	P		P	P	P	P	P	P	P
17	TCA2109021	AYUSH		P	P	P	P	P	P	P	P
18	TCA2109022	DEEPAK KUMAR	P		P	P	P	P	P	P	P
19	TCA2109023	DEEPANSHU	P	P		P	P	P	P	P	P
20	TCA2109025	DHRUV JAIN	P	P	P	P		P	P	P	P
21	TCA2109027	DIVYANSH POOTHIYA	P	P	P	P	P	P		P	P
22	TCA2109029	GOBIND SINGH	P	P	P	P	P	P	P	P	
23	TCA2109030	HARDIK JAIN	P	P	P	P	P	P	P	P	P
24	TCA2109031	HARSH AGARWAL	P	P	P	P	P	P	P	P	P
25	TCA2109032	HARSH VISHNOI	P	P	P	P	P	P	P	P	
26	TCA2109033	HARSH YADAV	P	P	P	P	P	P		P	P
27	TCA2109034	HARSHIKA JAIN	P	P	P	P		P	P	P	P
28	TCA2109035	HIMANSHU KUMAR	P	P		P	P	P	P	P	P
29	TCA2109036	HIMANSHU VERMA		P	P	P	P	P	P	P	P
30	TCA2109038	KHUSHI SAXENA	P	P	P	P		P	P	P	P
31	TCA2109039	KRATI JAIN	P	P	P	P	P	P		P	P
32	TCA2109040	KUMAR AAYUSH	P	P	P	P	P	P	P	P	
33	TCA2109042	MOHD FAISAL	P	P	P	P	P	P	P	P	P
34	TCA2109043	MOHD FRAIZ	P	P	P	P	P	P	P	P	P
35	TCA2109044	MOHD SAYEM RAZA KHAN	P	P	P	P	P	P	P	P	P
36	TCA2109045	NAMAN JAIN	P	P	P	P	P	P	P	P	
37	TCA2109046	NIPUR JAIN	P	P	P	P	P	P	P	P	P
38	TCA2109047	PALASH JAIN	P	P	P	P	P	P	P	P	P
39	TCA2109051	PRIYAM SIDDHU	P	P	P	P	P	P	P	P	P
40	TCA2109053	RITIK KUMAR	P	P	P	P	P	P	P	P	
41	TCA2109054	ROHIT JAIN	P	P	P	P	P	P	P		P
42	TCA2109055	ROHIT KUMAR	P	P		P	P	P	P	P	P
43	TCA2109056	SAMEEP AGARWAL		P	P	P	P	P	P	P	P
44	TCA2109057	SAMYAK JAIN	P	P	P	P		P	P	P	P

B.Tech IV (Fourth) Semester Examination 2023-24**Course Code:** ECS404**Paper ID:** 09624406**Software Engineering****Time:** 3

Hours

Max.**Marks:** 60**Note:** Attempt **all** questions.

Q1.	Answer the following (limit your answer to 50 words):	Marks	Category
a)	Explain software development life cycle.	2	Remembering & Understanding
	Or		
	What do you mean by Functional Dependency?	2	Remembering & Understanding
b)	Define Problem Partitioning concept in details.	2	Remembering & Understanding
	Or		
	Define the difference between Top-Down and Bottom up approach.	2	Remembering & Understanding
c)	What is Testing? List it's various types.	2	Remembering & Understanding
	Or		
	What is the role of Testing in the area of SDLC?	2	Remembering & Understanding
d)	State the role of Test Cycle.	2	Remembering & Understanding
	Or		
	Define various characteristics of Modern Tools and Automation in software engineering.	2	Remembering & Understanding
e)	Define Software Reliability and its importance in details.	2	Remembering & Understanding
	Or		
	Define your understanding about Computer Aided Software Engineering (CASE).	2	Remembering & Understanding
2.	Consider the following information about a university database: Professors have an SSN, a name, an age, a rank, and a research specialty.	10	Applying

	• Projects have a project number, a sponsor name (e.g., NSF), a starting date, an ending date, and a budget. • Graduate students have an SSN, a name, an age, and a degree program (e.g., M.S. or Ph.D.). • Each project is managed by one professor (known as the project's principal investigator). • Each project is worked on by one or more professors (known as the project's co-investigators). • Professors can manage and/or work on multiple projects. • Each project is worked on by one or more graduate students (known as the project's research assistants). • When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one. • Departments have a department number, a department name, and a main office. • Departments have a professor (known as the chairman) who runs the department. • Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job. • Graduate students have one major department in which they are working on their degree. • Each graduate student has another, more senior graduate student (known as a student advisor) who advises him or her on what courses to take. Design and draw an ER diagram that captures the information about the university. Show the cardinality ratio and participation as well.		
	Or		
	Consider the following set of requirements for a UNIVERSITY database that is used to keep track of students' transcripts. • The university keeps track of each student's name, student number, Social Security number, current address and phone number, permanent address and phone number, birth date, sex, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and ZIP Code of the student's permanent address and to the student's last name. Both Social Security number and student number have unique values for each student.	10	Applying

	- Each department is described by a name, department code, office number, office phone number, and college. Both name and code have unique values for each department. - Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of the course number is unique for each course. - Each section has an instructor, semester, year, course, and section number. The section number distinguishes sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester. - A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3, or 4). Draw an ER diagram for that schema. Specify key attributes of each entity type along with cardinality ratio and participation. You can make appropriate assumptions to make the specification complete.		
3.	Discuss the following briefly: a) Structured Approach b) Distributed Software Design	5+5=10	Remembering & Understanding
	Or		
	Write short notes on the following: a) UML b) Abstraction	5+5=10	Remembering & Understanding
4.	Explain in detail White Box Testing and Black Box Testing.	10	Remembering & Understanding
	Or		
	Discuss differences between Alpha and Beta Testing.	10	Remembering & Understanding
5.	Discuss differences between Static Testing Tool and Dynamic Testing Tool.	10	Remembering & Understanding
	Or		
	Write short notes on the following: a) Test Estimation b) Test Cases	5+5=10	Remembering & Understanding
6.	Write short notes on the following: a) Documentation b) Error Reporting	5+5=10	Remembering & Understanding
	Or		
	Discuss in detail about Updates-Upgrades-Patches-Versions. Why these changes are provided under maintenance of the software?	10	Remembering & Understanding

FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Year: 2nd Sem: 4th Branch: CSE Batch:
Software Engineering (ECS404)
A.Y.: 2023-24 EVEN SEM

COURSE FILE CLOSURE REPORT

Course file consists of the following details:

- Vision & Mission of Institute
- Vision & Mission of Department
- Program Educational Objectives (PEOs)
- Program Outcomes (POs)
- Program Specific Outcomes (PSOs)
- Institute Academic Calendar
- Course Evaluation Scheme
- Course Outcomes
- CO-PO-PSO Mapping
- Lecture Plan & Course Coverage
- Time Table-Faculty
- Time Table- Sections
- List of Assignments/Quizzes
- Class Test Paper (CT1, CT2, and CT3)
- Class Test Marks (CT1, CT2, and CT3)
- Class Test Solution
- Assignments/Quizzes Solution
- List of Slow Learner Students of Class test and action taken
- List of Fast Learner Students of Class test and action taken
- Previous Year Question Papers
- **End Semester Exam Analysis**
- Direct and Indirect CO Attainment- CTs & End Sem

Subject Teacher Name

Verified by IQAC

HOD