

Institute/ School Name	School of Engineering and Technology		
Department Name	Department of Computer Science & Engineering		
Program Name	Bachelor of Engineering (Computer Science & Engineering): B.E (CSE)		
Course Code	24CS015	Course Name	Object Oriented Software Engineering
L-T-P (Per Week)	3-0-2	Course Credits	04
Academic Year	2025-26	Semester/Batch	4 th /2024-2028
Pre-requisites (if any)	None		
NHEQF Level	5	SDGs	4,9
Course Coordinator	Dr. Praveen Kantha		

1. Scope and Objective of the Course:

This course introduces object-oriented software engineering principles for developing reliable and maintainable software systems. It covers object-oriented analysis, design, UML-based modelling, implementation, testing, and design patterns with a strong industry focus. Learners will develop skills in requirement analysis, system modelling, object-oriented design, software implementation, testing, quality assurance, problem-solving, and building scalable and maintainable software solutions.

2. Programme Outcomes (POs):

At the end of the programme, students will be able to:	
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 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	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 teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	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 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 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.

3. Course Learning Outcomes (CLO):

After completing the course, the students will be able to:

- CLO1:** Understand fundamental concepts, principles, and systematic processes involved in Software Engineering for developing and maintaining software systems.
- CLO2:** Analyze software-engineering models and select appropriate design and development techniques to create efficient, reliable, industry-relevant software solutions that enhance employability.
- CLO3:** Evaluate the benefits and limitations of UML as a standard modelling notation and select appropriate UML diagrams for real-time software development scenarios.
- CLO4:** Predict software application cost using appropriate software metrics and select suitable metrics to support effective project planning, resource optimization, and analytical decision-making skills.
- CLO5:** Apply software testing, project scheduling, risk management, and quality assurance techniques to predict potential software issues while ensuring software reliability through practical activities.

4. CLO-PO Mapping Matrix:

Course Learning Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	NHEQF Level Descriptor
CLO1	H	L										L	Q1
CLO2		M	H	L							L	L	Q2
CLO3	M		L		H	L							Q3,Q4
CLO4			H			L		M	L				Q2,Q3
CLO5					H		L		L				Q3

5. ERISE Grid Mapping:

Feature Enablement	Level (1-5, 5 being highest)
Entrepreneurship	2
Research/Innovation	1
Skills	5
Employability	4

6. Recommended Books (Reference Books/Text Books):

- B01:** Pressman, R. S., Software Engineering: a practitioner's approach (9th Edition). Pressman and Associates.
- B02:** Schach, S. R., Object-oriented and classical software engineering (6th Edition). McGraw-Hill.
- B03:** Jacobson, I., Object-oriented software engineering: A use case driven approach (7th Edition). Pearson Education India.
- B04:** Unni, H., Scrum for startups: Accelerating growth with agile practices (1st Edition). Notion Press.
- B05:** Mathur, A. P., Foundations of software testing, (2nd Edition). Pearson Education India.
- B06:** Humble, J., & Farley, D., Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (1st Edition). Addison-Wesley Professional.

7. Other readings and relevant websites:

Resources	Link of Journals, Magazines, Websites and Research Papers
R1	Link for UMLet, a free and open-source UML tool with a straightforward interface for quickly creating diagrams. http://www.umlet.com
R2	Online NPTEL courses on Software Engineering offered by IIT Bombay, featuring Professors Rushikesh K Joshi, Umesh Bellur, and N.L. Sarda. https://nptel.ac.in/courses/106101061

R3	Online Software Engineering courses offered by Coursera. https://in.coursera.org/courses?query=software%20engineering
R4	Perera, P., Perera, S., Jin, X., Rashidi, M., Nanayakkara, S., Yazbek, G., & Yazbek, A. (2025). An Innovative Software Development Methodology for Deep Learning-Driven Visual Computing in Built Environment Applications. <i>Journal of Information Technology in Construction (ITcon)</i> , 30(41), 1017-1040.
Resources	Link of Audio-Video resources
V1	https://www.youtube.com/watch?v=K9eN_4ob0ZE
V2	https://www.uml.org/resource-hub.htm
V3	https://nptel.ac.in/courses/106105087
V4	https://hp.chitkara.edu.in//listOnlineResources.php
V5	https://www.youtube.com/watch?v=h9K1NnqwUvE

* Resources uploaded on ERP system is accessible to all the students registered for the course.

8. Recommended Tools and Platforms:

- Lucidchart (<https://www.lucidchart.com>)
- Draw.io (<https://app.diagrams.net/>)
- Selenium(<https://www.selenium.dev/>)
- Unittest(<https://docs.python.org/3/library/unittest.html>)
- JUnit (<https://junit.org>)
- Bugzilla (<https://www.bugzilla.org/>)

9. Course Plan:

Lecture Number	Topics	Weightage in ETE (%)	Instructional Resources
1-2	Introduction to Software Engineering: The Evolving Role of Software, Changing nature of software		B01, V1, R1
3-6	The Software Process: Software Engineering—Layered Technology, Process Models: The Waterfall Model, Evolutionary Process Model: Incremental Models, Spiral Model		B01, B02, V2
7-10	Introduction to UML and modelling software, Introduction to Use-case Diagram and Class diagram Practice Problems: <ul style="list-style-type: none">• Draw Use-Case Diagram of Transport Management System• Draw Class Diagram of Transport Management System	20	Annexure-I
11-13	An Agile View of Process: what is agility, what is an agile process, Agile Process Models: extreme programming (XP), ASD, RAID, Kanban, Introduction to DevOps.		B01, B02, R1, B03, B05, V4,B06, V5
14-15	Scrum: Introduction to Scrum, Scrum Framework, Scrum Roles and Responsibilities, Scrum Artifacts, Tools and Techniques for Scrum		B01, B02, B03, B05, R2, V3
16-17	Requirements Engineering: Requirements Engineering Tasks: Initiating Requirement, Engineering Process, Eliciting Requirements		B01, B02, R3
18-20	Design Engineering: Design concepts and model: Cohesion, Coupling, Information hiding, Functional independence, Refinement, Data design, Architectural design Patterns: Layered, Client-Server, Microservices, Designing class-based components		B02, B05, R2, V1
21-24	Implementation & Configuration Management: Coding Standards, Best Practices, and Secure Coding, Version Control Systems: Git, GitHub/GitLab workflows, Branching Strategies and Code Reviews, Cloud-Native Application Development Overview, Containerization using Docker, Continuous Integration/Continuous Delivery (CI/CD) Pipelines	20	B01, B02, B06, V5

25-28	Introduction to Sequence Diagrams, and Activity Diagram Practice Problems: • Draw Sequence Diagram for Transport Management System • Draw Activity Diagram for Transport Management System		Annexure-I
29-30	Software Testing Strategies and Tactics: A strategic approach for Software Testing, Software Testing Strategies: Unit Testing, Integration Testing		B01, B02, BO5, R2, V3
31-34	Practice Problems: • Study of Test Management Tool “JMeter” • Design and Develop Test Cases for Unit Testing and Integration Testing in Recruitment System		Annexure-I
35-37	Validation Testing, System Testing, Test strategies for Object-Oriented Software- Unit Testing in the OO Context, Integration Testing in the OO Context	20	B01, B02, R2, V1
38-40	White-Box Testing Techniques: Basis Path Testing, Control Structure Testing: condition and loop testing, Black-Box Testing Techniques: Equivalence Partitioning and Boundary Value Analysis		B01, B02, B03, BO5, R2, V3
41-44	Practice Problems: • Study of Test Management Tool “Unittest” • Design and Develop Test Cases for White-Box and Black-Box Testing Techniques in a Recruitment System		Annexure-I
45-46	Testing Object Oriented Applications: Testing OOA and OOD model, Object Oriented Testing Strategies, Object Oriented Testing Methods, Introduction to Automation Testing		B01, B02, B04, BO5, R2, V3
47-50	Study Web testing tool “SELENIUM” and solve Practice Problems: • Cross-Browser Testing for Webpage Compatibility • Cross-Platform Testing on Web Application		Annexure-I
51-52	Project Management & Metrics: The management spectrum, Metrics for process & project, Metrics for Software Quality, Estimation		B01, B02, B04, BO5, R2, V1
53-54	Product Metrics: Metrics for the requirement model, Metrics for the design model, Metrics for testing	20	B01, B02, BO5, R2, V3
55-58	Introduction to State chart diagrams, Data Flow diagram Practice Problems: • Draw State Chart Diagram for Transport Management System • Draw Data Flow Diagram for Transport Management System		Annexure-I
59-60	Software Project Planning: Objective, Software Scope and Resources, Software Project Estimation and Decomposition Techniques (LOC, FP)		B01, B02, BO5, R2, V1
61-62	Empirical Estimation Models: COCOMO Model, Estimation of Object-Oriented Projects		B01, B02, BO5, R2, V1
63-64	Project Scheduling: Basic concepts of scheduling, Project Scheduling, Earned Value Analysis		B01, B02, BO5, R2, V2
65-68	Practice Problems: Develop a Gantt chart, and PERT chart for the Examination System		Annexure-I
69-70	Risk Management: Software Risks & Risk Strategies, Risk Identification, Risk Projection, Risk Mitigation, Monitoring and Management (RMMM) plan	20	B01, B02, B03, B04, BO5, R2, V1
71-72	Practice Problems: Study of Bug Tracking Tool “BUGZILLA”		Annexure-I
73-75	Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, and the ISO 9000 quality. Standards, Capability Maturity Model (CMM)		B01, B02, BO5, R2, V3

10. Industry Interventions:

Industry curated course “Software Engineering Fundamentals-Software Development and Testing” at the link below:

- https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01384297011411353628269_shared/overview

11. Innovative Pedagogies:

- Case study-based learning (Annexure-II)

12. Action plan for different types of learners

Slow Learners	Average Learners	Advanced Learners
Remedial Classes	Assignments (Annexure-III)	Industry Certification (Annexure-IV)

13. Evaluation Scheme & Components:

Evaluation Component	Type of Component	No. of Assessments	Weightage of Component	Mode of Assessment (Offline / Online)
Internal Component 1	Formative Assessments (FAs)	01*	20%	Offline
Internal Component 2	Sessional Tests (STs)	03**	30%	
External Component	End Term Examination	01	50%	
Total				100%

* Formative Assessments (FA) is mandatory.

** Out of 03 STs, the best 02 STs marks will be considered to evaluate final marks.

14. Details of Evaluation Components:

Evaluation Component	Description	Syllabus Covered (%)	Timeline of Examination	Weightage (%)
Internal Component 1	FA1	• Industry Certification • Lab Assessments	Will be intimated one week prior	20%
Internal Component 2	ST 01	Up to 40% (Lectures 1-30)		30%
	ST 02	41% - 80% (Lectures 31-60)		
	ST 03	100% (Lectures 1-75)		
External Component	End Term Examination*	100%		50%
Total				100

* Minimum 75% attendance is required to become eligible for appearing in the End Semester Examination

15. Format of Evaluation Components:

Type of Assessment	Total Marks	Industry Certification	Hands-on Assessments	1 Mark MCQ	2 Marks	5 Marks	10 Marks
Formative Assessments	20	5	15	-	-	-	-
Sessional Tests	40	-	-	5	5	3	1
End Term Examination	60	-	-	5	5	5	2

16. Revision (if any):

Academic Year of Previous Version	2024-2025	Percentage of Revision	11%
Topics Added/Removed:			
<ul style="list-style-type: none"> User interface analysis and design, Interface analysis, and Interface design steps, Building Analysis Model: Requirement Analysis, Data modeling Concepts, Flow Oriented Modelling, Junit, Practice Problems: Study of Test Management Tool “TestLink”, Design and Develop Test Cases for Unit Testing and Integration Testing Techniques in an E-Ticketing System (Removed) Kanban, Introduction to DevOps, Unittest, Implementation & Configuration Management: Coding Standards, Best Practices, and Secure Coding, Version Control Systems: Git, GitHub/GitLab workflows, Branching Strategies and Code Reviews, Cloud-Native Application Development Overview, Containerization using Docker, Continuous Integration/Continuous Delivery (CI/CD) Pipelines, Practice Problems: Study of Test Management Tool “JMeter”, Design and Develop Test Cases for Unit Testing and Integration Testing in Recruitment System (Added) 			

17. This Document is:

Designation	Name	Signature
Prepared by Course Coordinator	Dr. Praveen Kantha	P.Kantha
Verified by Assistant Dean	Dr. Hakam Singh	H.Singh
Date	12/01/2026	

Annexure-I

S. No	Experiment
1.	Draw Use-Case Diagram of Transport Management System
2.	Draw Class Diagram of Transport Management System
3.	Draw Sequence Diagrams for Transport Management System
4.	Draw Activity Diagram for Transport Management System
5.	Study of Test Management Tool “JMeter”
6.	Design and Develop Test Cases for Unit Testing and Integration Testing in Recruitment System
7.	Study of Test Management Tool “Unittest”
8.	Design and Develop Test Cases for White-Box and Black-Box Testing Techniques in a Recruitment System
9.	Cross-Browser Testing for Webpage Compatibility using Web testing tool “SELENIUM”
10.	Draw State Chart Diagram for Transport Management System
11.	Draw Data Flow Diagram for Transport Management System
12.	Develop a Gantt chart, and PERT chart for the Examination System
13.	Study of Bug Tracking Tool “BUGZILLA”

Annexure-II

S. No	Topics
1	Case study-based learning: Software Requirement Specification (SRS) Document

Description:
The students will be introduced to a university-based case study involving departments such as Infrastructure, Examination, Transport, and the Registrar's Office. They will analyze departmental processes, identify automation requirements, and create a detailed Software Requirements Specification (SRS) document to address the identified challenges.

Key Objectives:

- To analyze departmental processes and identify potential areas for automation within university operations.
- To develop a detailed Software Requirements Specification (SRS) document addressing the identified issues.

Group Formation Requirements:

- Each team should consist of a maximum of 10 members.
- Each group must appoint a student coordinator to lead and facilitate discussions and activities.

Annexure-III

S. No	Topics
1	<p>Assignment 1: UML Modeling and Use-Case Diagram</p> <p>Description:</p> <p>Analyze a given software system (e.g., Examination Management System or Online Shopping System), identify functional requirements, and create a Use-Case diagram along with actors and interactions. Include at least 5 use cases and describe each scenario.</p> <p>Key Objectives:</p> <ul style="list-style-type: none"> • To understand requirements elicitation and modeling using UML. • To practice creating Use-Case diagrams and documenting system interactions. <p>Assignment 2: Microservices Design and API Documentation</p> <p>Description:</p> <p>Design a small microservices-based application (e.g., Online Movie Ticket Booking System) with at least 3 independent services. Create service interaction diagrams and document REST APIs for each service, including endpoints, request/response formats, and HTTP methods.</p> <p>Key Objectives:</p> <ul style="list-style-type: none"> • To understand microservices architecture and service decomposition. • To practice API design and documenting service interactions.

Annexure-IV

S. No	Topics
1	<p>Software Testing Management Certification Course</p> <p>Description:</p> <p>Advanced learners will be offered the Software Testing Management Certification Course through Infosys Springboard. The course introduces learners to software testing principles, strategies, and tools, focusing on managing testing activities effectively in software projects.</p> <p>Key Objectives:</p> <ul style="list-style-type: none"> • To understand the fundamentals of software testing, including types, levels, and testing life cycles. • To plan, execute, and manage testing activities using industry-standard methodologies and tools. • To design and implement test strategies, manage defects, and ensure software quality throughout the development process. <p>Online Course Link:</p> <p>https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0138417511142195202777_shared/overview</p>