# Course Plan

**Course Handout**

| Institute/School Name | Chitkara University Institute of Engineering & Technology | | |
|---|---|---|---|
| Department Name | Department of Computer Science and Engineering | | |
| Programme Name | B.E (Computer Science & Engineering) | | |
| Course Name | Object Oriented Software Engineering | Session | July – Dec 2025 |
| Course Code | 24CSE0210 | Semester/Batch | 3rd /2024 |
| L-T-P(Per Week) | 3-0-0 | Course Credits | 03 |
| Pre-requisite | Basic knowledge of object-oriented concepts. | NHEQF Level | 05[1] |
| Course Coordinator | Dr. Chetna | SDG Number | 1,3,9[4] |

## 1. Objectives of the Course

Software Engineering (SE) comprises the core principles consistent in software construction and maintenance: fundamental software processes and life cycles, mathematical foundations of software engineering, requirements analysis, software engineering methodologies and software quality frameworks and validation, software development, and maintenance environments and tools. In this course, an introduction to object-oriented software development process and design. Topics include iterative development, interpretation of requirements and use case documents into code; application of design notation in UML and use of commonly-used design patterns. The objectives of the course are:

i. to build an understanding of the software process models such as the waterfall and evolutionary models based on the software requirements and the SRS documents.
ii. to inculcate the skill in the application of computing-based solutions to societal and organizational problems.
iii. to develop, implement and manage quality control and how to ensure good quality software.
iv. to demonstrate ethical principles in the application of computing-based solutions to societal and organizational problems.

## 2. Course Learning Outcomes (CLOs)

After completion of the course, the student should be able to:

| | **Course Learning Outcome** | **\*POs** | **NHEQF Level Descriptor[1]** | **Sessions** |
|---|---|---|---|---|
| **CLO01** | Define the current theories, models, and techniques that provide a basis for the software lifecycle. | PO1, PO2, PO3, PO5,PO12 | Q1, Q2, Q3 | **9** |
| **CLO02** | Explain the use of techniques and tools necessary for engineering practice in one or more significant application domains. | PO3, PO4, PO5 | Q1, Q2, Q3 | **8** |
| **CLO03** | Apply the software engineering lifecycle by demonstrating competence in communication, planning, analysis, design, construction, and deployment. | PO1, PO2, PO3, PO4,PO5, PO7, PO11 | Q3, Q4 | **9** |
| **CLO04** | Examine strong communication and interpersonal skills, as well as professional and ethical principles when functioning as members and leaders of multi-disciplinary teams. | PO3, PO4, PO5 | Q4, Q5 | **9** |
| **CLO05** | Design software based on requirements analysis, verification, validation and to develop solutions to modern problems such as security, data science, and systems engineering. | PO1, PO2, PO3, PO5,PO6, PO7, PO11 | Q1, Q2, Q3, Q4, Q5, Q6 | **10** |
| | **Total Contact Hours** | | | **45** |

Revised Bloom's Taxonomy Terminology
\* PO's available at (shorturl.at/cryzF)
\*\*Cognitive Level =CL
\*\*\*Knowledge Categories = KC

---

[1] NHEQF Level Descriptor, Refer to Annexure & Learning outcomes descriptors for qualification for all levels on the NHEQF
[3]Types of Assessments can be referred from Type of Assessments. Refer to Annexure.
[4]For SDG Mapping with Courses, Pl refer SDG Mapping policy for Courses

**CLO-PO Mapping**

| Course Learning Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | Type of Assessment's[3] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLO01 | H | H | M | | H | | | | | M | | H | M | Summative |
| CLO02 | | H | H | H | | | | | | | | M | H | Summative |
| CLO03 | H | H | M | M | H | | M | | | | H | H | | Summative |
| CLO04 | | | H | H | M | | | | | | | M | H | Summative |
| CLO05 | H | H | H | | M | M | M | | | | M | | H | Summative |

H=High, M=Medium, L=Low

3. **Recommended Books:**

   **B01:** An Integrated Approach to Software Engineering by Pankaj Jalote, Narosa Publications, New Delhi, 2nd Edition,1997.

   **B02:** Software Engineering - A Practitioner's Approach, Roger S. Pressman, MGH, New Delhi. 5th Edition,2001.

   **B03:** Fundamentals of Software Engineering, Rajib Mall, PHI, New Delhi, 4th Edition, 2014.

   **B04:** Agile Software Development, Torgeir Dingsoyr, Tore Dyba, Nils Brede Moe, Springer, Jan,2010.

   **Reference Books:**

   **B05:** Testing and Quality Assurance for Component-based Software, by Gao, Tsao and Wu Artech HousePublishers,2003

   **B06:** Software Engineering by Ian Summerville, Pearson Education, 5th Edition, New Delhi,2014**.**

   **B07:** Handbook of Software Quality Assurance, by G. Gordon Schulmeyer, International Thomson Computer Press, 4th Edition, 2016

   **E-Resources:**
   https://library.chitkara.edu.in/subscribed-books.php

4. **Other readings and relevant websites:**

| Serial No | Link of Journals, Magazines, websites and Research Papers |
|---|---|
| 1. | https://onlinecourses.nptel.ac.in/noc20_cs68/preview |
| 2. | https://archive.nptel.ac.in/courses/106/105/106105182/ |
| 3. | https://nptel.ac.in/courses/106101061 |
| 4. | https://www.coursera.org/specializations/software-engineering |
| 5. | https://nptel.ac.in/courses/106105087 |
| 6. | https://www.coursera.org/learn/software-engineering-software-design-and-project-management |
| 7. | https://onlinecourses.nptel.ac.in/noc22_cs39/preview |
| 8. | https://se-iitkgp.vlabs.ac.in/ |

5. **Recommended Tools and Platforms**

https://app.smartdraw.com/?nsu=1
https://www.figma.com/files/team/1443870582755368449/recents-and-sharing? fuid=_1443870580934668847

6. **Course Plan**

| Lecture Number | Topics | Textbook |
|---|---|---|
| 1 | Detail Discussion of Course Handout (CHO), Introduction to Software Engineering: The Evolving Role ofSoftware, Changing nature of software | B01-Chapter-1 |
| 2 | The Software Process: Software Engineering–Layered Technology, Process Models: The Waterfall Model | B01-Chapter-2 |
| 3-4 | Evolutionary Process Models, Incremental Models, SpiralModel | B01-Chapter-2 |
| 5-7 | An Agile View of Process: what is agility, what is agile process, Agile Process Models: extreme programming (XP),ASD, Scrum, Introduction to UML and modelling software | B01-Chapter-2 B02-Chapter-2 B04-Chapter-1 |
| 8-9 | Requirements Engineering: Requirements Engineering Tasks: Initiating Requirement, Engineering Process, Eliciting Requirements | B02- Chapter-5 |
| 10-12 | Introduction to Use-case Diagram, Use-case Diagram of College Information System/Library Management system/ Hospital Management System/ Online shopping system/ Banking System | B02-Chapter-6 |
| 13 | Building Analysis Model: Requirement Analysis, Datamodelling Concepts, Flow Oriented Modelling | B02-Chapter-5 |

| | | |
|---|---|---|
| **ST-1(Lecture 1-13)** | | |
| 14-16 | Design Engineering: Design concepts and model, Data design, Architectural design, designing class-based components, User interface analysis and design, Interface analysis and Interface design steps | B01-Chapter-5 |
| 17-19 | Introduction to Class diagram, Class diagram for College Information System/Library Management system/ HospitalManagement System/ Online shopping system/Banking System | B03-Chapter-7 |
| 20 | Software Testing Strategies and Tactics: A strategic approach for Software Testing, Software Testing Strategies: Unit Testing | B01-Chapter-9 |
| 21-24 | Integration Testing, Validation Testing, System Testing, OOA and OOD model, Object Oriented Testing Strategies | B01-Chapter-9, B02-Chapter-19 |
| 25-26 | White-Box Testing Techniques: Basis Path Testing, ControlStructure Testing: condition and loop testing | B01-Chapter-9 B03-Chapter-10 |
| 27 | Black-Box Testing Techniques: Equivalence Partitioning andBoundary Value Analysis | B01-Chapter-9 B03-Chapter-10 |
| **ST-2 (Lecture 1-27)** **(ST1 syllabus also included)** | | |
| 28 | Introduction to Interaction diagrams, Draw interactive diagram for college information system/Library Managementsystem/ Hospital Management System/ Online shopping system/Banking System | B03-Chapter-7 |
| 29-30 | Project Management & Metrics: The management spectrum, Metrics for process & project, Metrics for Software Quality, Estimation. | B01-Chapter-2 |
| 31-32 | Product Metrics: Metrics for the requirement model, Metrics for the design model, Metrics for testing | B01-Chapter-2 B02-Chapter-23 |
| 33-34 | Introduction to Activity diagram, Activity diagram for college information system /Library Management system/ Hospital Management System/ Online shopping system/Banking System/Bug Removal | B03-Chapter-7 |
| 35 | Software Project Planning: Objective, Software Scope and Resources, Software Project Estimation and DecompositionTechniques (LOC, FP) | B03-Chapter-3 B02-Chapter-4 |
| 36-38 | Empirical Estimation Models: COCOMO Model, Estimation of Object-Oriented Projects | B01-Chapter-4 B02-Chapter-26 |
| 39-40 | Project Scheduling: Basic concepts of scheduling, Project Scheduling, Earned Value Analysis | B01-Chapter-4 B02-Chapter-27 |
| 41-43 | Risk Management: Software Risks & Risk Strategies, Risk Identification, Risk Projection, Risk Mitigation, Monitoring and Management (RMMM) plan | B01-Chapter-9 B03-Chapter-3 |
| 44-45 | Overview of Quality Management and Change Management | B03-Chapter-11 |
| **Sessional Test-3** | | |
| **End Term Examination (ETE-Complete Syllabus)** | | |

7. **Delivery/Instructional Resources Theory Plan:**

| Lect. No. | Topics | CLO | Book No, CH No, Page No | TLM[2] | ALM[3] | Web References | Audio-Video |
|---|---|---|---|---|---|---|---|
| 1 | Detail Discussion of Course Handout (CHO), Introduction to Software Engineering: The Evolving Role of Software, Changing nature of software | 1 | B01-Chapter-1 | Lecture Discussion | Assignment | https://www.geeksforgeeks.org/software-engineering/#int | https://youtu.be/AN5I6fFxyfs |
| 2 | The Software Process: Software Engineering–Layered Technology, Process Models: The Waterfall Model | 1, 2 | B01-Chapter-2 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering/#sdl | https://www.youtube.com/watch?v=uJpQlyT_CK4&list=PLxCzCOWd7aiEed7SKZBnC6ypFDWYLRvB2 |
| 3-4 | Evolutionary Process Models, Incremental Models, Spiral Model | 1, 2 | B01-Chapter-2 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering/#sdl | https://www.youtube.com/watch?v=uJpQlyT_CK4&list=PLxCzCOWd7aiEed7SKZBnC6ypFDWYLRvB2 |
| 5-7 | An Agile View of Process: what is agility, what is agile process, Agile Process Models: extreme programming (XP), ASD, Scrum, Introduction to UML and modelling software | 2 | B01-Chapter-2 B02-Chapter-2, B04-Chapter-1 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering-agile-development-models/ | https://www.youtube.com/watch?v=Xs6E-MAJbfE |
| 8-9 | Requirements Engineering: Requirements Engineering Tasks: Initiating Requirement, Engineering Process, Eliciting Requirements | 2,3 | B02- Chapter-5 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering/#sr | https://youtu.be/wEr6mwqu PLY |
| 10-12 | Introduction to Use-case Diagram, Use-case Diagram of College Information System/Library Management system/ Hospital Management System/ Online shopping system/Banking System | 2,3 | B02-Chapter-6 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/use-case-diagram/ | https://www.youtube.com/watch?v=HVSxE296JLc |
| 13 | Building Analysis Model: Requirement Analysis, Data modelling Concepts, Flow Oriented Modelling | 3 | B02-Chapter-5 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/analysis-modelling-in-software-engineering/ | https://www.youtube.com/watch?v=EhmHFtwFoQ0 |
| 14-16 | Design Engineering: Design concepts and model, Data design, Architectural design, designing class-based components, User interface analysis and design, Interface analysis and Interface | 3,4 | B01-Chapter-5 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering/#std | https://youtu.be/Q50ZyydS7 pI |
| 17-19 | Introduction to Class diagram, Class diagram for College Information System/Library Management system/ Hospital Management System/ Online shopping system/Banking System | 3,4 | B03-Chapter-7 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering-white-box-testing/ | https://youtu.be/Q50ZyydS7 pI |
| 20 | Software Testing Strategies and Tactics: A strategic approach for Software Testing, Software Testing Strategies: Unit Testing | 4 | B01-Chapter-9 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering-white-box-testing/ | https://youtu.be/Q50ZyydS7 pI |
| 21-24 | Integration Testing, Validation Testing, System Testing, Testing OOA and OOD model, Object Oriented Testing Strategies | 4 | B01-Chapter-9, B02-Chapter-19 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering-black-box- | https://youtu.be/PXYqu- OcBoY |

---

[2] Teaching Learning Methods
[3] Active Learning Methods

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | testing/ |
| 25-26 | White-Box Testing Techniques: Basis Path Testing, Control Structure Testing: condition and loop testing | 4 | B01-Chapter-9 B03-Chapter-10 | Lecture Discussion | Quiz Challenge Assignment | https://www.tutorialspoint.com/uml/uml_interaction_diagram.htm | https://www.youtube.com/watch?v=HVSxE296JLc |
| 27 | Black-Box Testing Techniques: Equivalence Partitioning and Boundary Value Analysis | 4 | B01-Chapter-9 B03-Chapter-10 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering/#spm | https://youtu.be/UEEWbe1Fo KI https://youtu.be/KqDlDubS- OU |
| 28 | Introduction to Interaction diagrams, Draw interactive diagram for college information system/Library Management system/ Hospital Management System/ Online shopping system/Banking System | 5 | B03-Chapter-7 | Lecture Discussion | Quiz Challenge Assignment | https://www.tutorialspoint.com/uml/uml_interaction_diagram.htm | https://www.youtube.com/watch?v=HVSxE296JLc |
| 29-30 | Project Management & Metrics: The management spectrum, Metrics for process & project, Metrics for Software Quality, Estimation. | 5 | B01-Chapter-2 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering/#spm | https://youtu.be/UEEWbe1Fo KI https://youtu.be/KqDlDubS- OU |
| 31-32 | Product Metrics: Metrics for the requirement model, Metrics for the design model, Metrics for testing | 5 | B01-Chapter-2, B02-Chapter-23 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering/#spm | https://www.youtube.com/watch?v=jiOaywcVssQ |
| 33-34 | Introduction to Activity diagram, Activity diagram for college information system /Library Management system/ Hospital Management System/ Online shopping system/Banking System/Bug Removal | 5 | B03-Chapter-7 | Lecture Discussion | Quiz Challenge Assignment | https://se-iitkgp.vlabs.ac.in/exp/statechart-activity-modeling/statechart-activity.html | https://www.youtube.com/watch?v=LyhTDsjjjrE |
| 35 | Software Project Planning: Objective, Software Scope and Resources, Software Project Estimation and Decomposition Techniques (LOC, FP) | 5,6 | B03-Chapter-3 B02-Chapter-4 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering-role-and-responsibilities-of- a-software-project-manager/ | https://youtu.be/5pwc2DYlK QU |
| 36-38 | Empirical Estimation Models: COCOMO Model, Estimation of Object-Oriented Projects | 5,6 | B01-Chapter-4, B02-Chapter-26 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering-cocomo-model/ | https://youtu.be/D04uxZpgp 6M |
| 39-40 | Project Scheduling: Basic concepts of scheduling, Project Scheduling, Earned Value Analysis | 5,6 | B01-Chapter-4 B02-Chapter-27 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/software-engineering-capability-maturity-model- cmm/ | https://youtu.be/NGmPTl6Ad o0 |
| 41-43 | Risk Management: Software Risks & Risk Strategies, Risk Identification, Risk Projection, Risk Mitigation, Monitoring and Management (RMMM) plan | 5,6 | B01-Chapter-9 B03-Chapter-3 | Lecture Discussion | Quiz Challenge Assignment | https://www.geeksforgeeks.org/integrating-risk-management-in-sdlc-set-1/ | https://youtu.be/IBL9MqvpPl M |
| 44-45 | Overview of Quality Management and Change Management | 6 | B03-Chapter-11 | Lecture Discussion | Quiz Challenge Assignment | https://www.tutorialspoint.com/software_testing_dictionary/quality_management.htm | https://youtu.be/3MgEkS8_jz o |

**Remedial Classes[4]**

| Slow Learners | Average Learners | Fast Learners |
|---|---|---|
| • Remedial Classes on Saturdays<br>• Encouragement for improvement using Peer Tutoring<br>• Use of Audio and Visual Materials<br>• Use of Real-Life Examples | • Workshops<br>• Formative Exercises used to highlight concepts and notions<br>• E-notes and E-exercises to read ahead of the pedagogic material. | • Engaging students to hold hands of slow learners by creating a Peer Tutoring Group<br>• Design solutions for complex problems<br>• Design solutions for complex problems<br>• Presentation on topics beyond those covered in CHO |

8. **Self-Learning[5]**

Assignments to promote self-learning, survey of contents from multiple sources. During this project, various challenges and situations will be encountered that require the students to learn new tools, technologies, and soft skills outside of our coursework. Some of the key self-learning outcomes include:

1. **New Programming Languages/Frameworks:** They will learn how to work with [e.g., ReactJS, Node.js] which were not part of their curriculum but essential for their project, they explored API integration, asynchronous operations, and real-time data updates.

2. **Version Control with Git & GitHub:** The students will become proficient in using Git for source code management and will learn how to resolve and merge conflicts and manage project collaboration via GitHub.

3. **Software Development Methodologies**: The students will gain hands-on experience with Agile development and the Scrum process. They will practice creating sprints, daily stand-ups, and iterative development.

4. **Testing & Debugging:** The students will learn how to write unit tests and conduct system testing, used tools like Postman, Selenium, or JUnit for testing purposes.

5. **UI/UX Design:** The students will study design principles to build intuitive and user-friendly interfaces. The students will also use tools such as Figma or Adobe XD for prototyping.

6. **Time Management and Team Collaboration:** Will Improve their ability to manage time and meet deadlines under pressure. The students will use Trello/Notion/MS Teams for task tracking and effective communication.

7. **Deployment & Hosting:** The students will gain knowledge of how to deploy their application on platforms such as AWS. They will also understand concepts like domain linking, database hosting, and backend deployment.

8. **Soft Skills:** Improved their presentation skills, documentation, and ability to explain technical concepts clearly. The students will also learn how to handle constructive criticism during peer reviews and presentations.

9. **Delivery Details of Content Beyond Syllabus[6]**

Content beyond the syllabus covered (if any) should be delivered to all students that would be planned, and schedule notified accordingly. Different UML diagrams will be made by the students. Cyber security trends are also coming up in software engineering these days. Cybersecurity has become a critical focus in software engineering due to increasing threats and data breaches. Trends like Secure Software Development Life Cycle (SSDLC), DevSecOps, and AI-driven threat detection are being integrated to build more resilient and secure applications from the ground up.

| S.No | Advanced Topics, Additional Reading, Research papers and any | CLO | POs | ALM | References/MOOCS |
|---|---|---|---|---|---|
| 1 | Practicing different diagrams and learning latest trends in Software Engineering | NA | 1, 3, 4, 5, 10 | Discussions and Quiz | https://www.youtube.com/watch?v=fFbZv894at0 |

---

[4] Refer to Annexure
[5] Refer to Annexure
[6] Refer to Annexure

**10.** **Evaluation Scheme & Components:**

| Assessment Type[7] | Evaluation Component[8] | Type of Component[9] | No. of Assessments[10] | % Weightage of Component | Max. Marks | Mode of Assessment | CLO |
|---|---|---|---|---|---|---|---|
| Summative | Component 1 | ST-1 (Demonstration) | 01** | 10% | 10 | Online | 1 - 2 |
| Summative | Component 2 | ST-2 (MCQ-Based Evaluation) | 01** | 10% | 10 | Online | 1 - 4 |
| Summative | Component 3 | ST-3 (Project-Based Evaluation) | 01** | 20% | 20 | Offline | 1 - 5 |
| Summative | Component 4 | End Term Examination | 01*** | 60% | 60 | Online | 1 - 5 |

** Students will have to appear in all Sessional Tests.
** Makeup Examination will compensate for either ST-1 or ST-2 (Only for genuine cases, based on the Dean's approval).
** Makeup Examination will be in the form of ST-1 (Demonstration) only after Dean's approval
** ST-3 will be a project-based evaluation (No makeup exam will be taken for ST-3)
 *** As per Academic Guidelines, a minimum of 75% attendance is required to become eligible for appearing in the End Semester Examination.

**11.** **Syllabus of the Course:**

| S.No. | Topic (s) | No. of Sessions | Weightage % |
|---|---|---|---|
| 1 | Introduction to Software Engineering: The Evolving Role of Software, changing the nature of software <br> The Software Process: Software Engineering–Layered Technology, Process Models: The Waterfall Model <br> Evolutionary Process Models, Incremental Models, Spiral Model <br> An Agile View of Process: what is agility, what is agile process, Agile Process Models: extreme programming (XP), ASD, Scrum, Introduction to UML and modelling software <br> Requirements Engineering: Requirements Engineering Tasks: Initiating Requirement, Engineering Process, Eliciting Requirements <br> Introduction to Use-case Diagram, Use-case Diagram of College Information System/Library Management system/ Hospital Management System/ Online shopping system/ Banking System <br> Building Analysis Model: Requirement Analysis, Data modelling Concepts, Flow Oriented Modelling | 13 | 28% |
| | **Sessional Test-1(Demo Based Evaluation)** | | |
| 2 | Introduction to Software Engineering: The Evolving Role of Software, changing the nature of software <br> The Software Process: Software Engineering–Layered Technology, <br> Process Models: The Waterfall Model <br> Evolutionary Process Models, Incremental Models, Spiral Model <br> An Agile View of Process: what is agility, what is agile process, Agile Process Models: extreme programming (XP), ASD, Scrum, Introduction to UML and modelling software <br> Requirements Engineering: Requirements Engineering Tasks: Initiating Requirement, Engineering Process, Eliciting Requirements <br> Introduction to Use-case Diagram, Use-case Diagram of College Information System/Library Management system/ Hospital Management System/ Online shopping system/ Banking System <br> Building Analysis Model: Requirement Analysis, Data modelling Concepts, Flow Oriented Modelling <br> Design Engineering: Design concepts and model, Data design, Architectural design, designing class-based components, User interface analysis and design, Interface analysis and Interface design steps <br> Introduction to Class diagram, Class diagram for College Information System/Library Management system/ Hospital Management System/ Online shopping system/Banking System <br> Software Testing Strategies and Tactics: A strategic approach <br> for Software Testing, Software Testing Strategies: Unit Testing <br> Integration Testing, Validation Testing, System Testing, Test <br> strategies for Object Oriented Software- Unit Testing in the OO Context, Integration Testing in the OO Context <br> White-Box Testing Techniques: Basis Path Testing, Control Structure Testing: condition and loop testing | 27 | 60% |

---

[7] Refer to Annexure 2 of NCrF
[8] Refer to Annexure
[9] Refer to Annexure
[10] Refer to Annexure

24CSE0210/ Object Oriented Software Engineering

| | | Black-Box Testing Techniques: Equivalence Partitioning and Boundary Value Analysis Testing Object Oriented Applications: Testing OOA and OOD model, Object Oriented Testing Strategies, Object Oriented Testing Methods | | |
|---|---|---|---|---|
| | | **Sessional Test -2 (ST1 syllabus also included)** | | |
| | 3 | Introduction to Software Engineering: The Evolving Role of Software, Changing nature of software<br>The Software Process: Software Engineering–Layered Technology, Process Models: The Waterfall Model<br>Evolutionary Process Models, Incremental Models, Spiral Model<br>An Agile View of Process: what is agility, what is agile process, Agile Process Models: extreme programming (XP), ASD, Scrum, Introduction to UML and modelling software<br>Requirements Engineering: Requirements Engineering Tasks: Initiating Requirement, Engineering Process, Eliciting Requirements<br>Introduction to Use-case Diagram, Use-case Diagram of College Information System/Library Management system/ Hospital Management System/ Online shopping system/ Banking System<br>Building Analysis Model: Requirement Analysis, Data modelling Concepts, Flow Oriented Modelling<br>Design Engineering: Design concepts and model, Data design, Architectural design, designing class-based components, User interface analysis and design, Interface analysis and Interface design steps<br>Introduction to Class diagram, Class diagram for College Information System/Library Management system/ Hospital Management System/ Online shopping system/Banking System<br>Software Testing Strategies and Tactics: A strategic approach<br>for Software Testing, Software Testing Strategies: Unit Testing<br>Integration Testing, Validation Testing, System Testing, OOA and OOD model, Object Oriented Testing Strategies<br>White-Box Testing Techniques: Basis Path Testing, Control Structure Testing: condition and loop testing<br>Black-Box Testing Techniques: Equivalence Partitioning and Boundary Value Analysis<br>Introduction to Interaction diagrams, Draw interactive diagram for college information system/Library Management system/ Hospital Management System/ Online shopping system/Banking System<br>Project Management & Metrics: The management spectrum, Metrics for process & project, Metrics for Software Quality, Estimation.<br>Product Metrics: Metrics for the requirement model, Metrics<br>for the design model, Metrics for testing<br>Introduction to Activity diagram, Activity diagram for college information system /Library Management system/ Hospital Management System/ Online shopping system/Banking System/Bug Removal<br>Software Project Planning: Objective, Software Scope and<br>Resources, Software Project Estimation and Decomposition Techniques (LOC, FP)<br>Empirical Estimation Models: COCOMO Model, Estimation of Object-Oriented Projects<br>Project Scheduling: Basic concepts of scheduling, Project Scheduling, Earned Value Analysis<br>Risk Management: Software Risks & Risk Strategies, Risk Identification, Risk Projection, Risk Mitigation, Monitoring and Management (RMMM) plan<br>Overview of Quality Management and Change Management | 45 | **100%** |
| | | **Sessional Test-3** | | |
| | | **End Term Examination (ETE-Complete Syllabus)** | | |

12. **Academic Integrity Policy:**

Education at Chitkara University builds on the principle that excellence requires freedom where Honesty and integrity are its prerequisites. Academic honesty in the advancement of knowledge requires that all students and Faculty respect the integrity of one another's work and recognize the importance of acknowledging and safeguarding intellectual property. Any breach of the same will be tantamount to severe academic penalties.

**This Document is approved by:**

| Designation | Name | Signature |
|---|---|---|
| **Course Coordinator** | Dr. Chetna | |
| **Head-Academic Delivery** | Dr. Mrinal Paliwal | |
| **Dean** | Dr. Rishu Chhabra | |
| **Date(DD/MM/YYYY)** | 25.06.2025 | |

## Annexure

1.  **Pre- requisite**
    Mention The Pre-requisite skill set or course/s if it is expected to be studies before this course, otherwise write "not applicable".

2.  **NHEQF levels**
The NHEQF levels represent a series of sequential stages expressed in terms of a range of learning outcomes against which typical qualifications are positioned/located. NHEQF level 4.5 represents learning outcomes appropriate to the first year (first two semesters) of the undergraduate programme of study, while Level 8 represents learning outcomes appropriate to the doctoral-level programme of study.

Table 1: Higher education qualifications at different levels on the NHEQF

| NHEQF level | Examples of higher education qualifications located within each level |
|---|---|
| Level 4.5 | Undergraduate Certificate. Programme duration: First year (first two semesters) of theundergraduate programme, followed by an exit 4-credit skills-enhancement course(s). |
| Level 5 | Undergraduate Diploma. Programme duration: First two years (first four semesters) of the undergraduate programme, followed by an exit 4-credit skills-enhancement course(s) lasting two months. |
| Level 5.5 | Bachelor's Degree. Programme duration: First three years (Six semesters) of the four-yearundergraduate programme. |
| Level 6 | Bachelor's Degree (Honours/ Honours with Research). Programme duration: Four years(eight semesters). |
| Level 6 | Post-Graduate Diploma. Programme duration: One year (two semesters) for those whoexit after successful completion of the first year (two semesters) of the 2-year master's programme. |
| Level 6.5 | Master's degree. (e.g. M.A., M.Com., M.Sc., etc.) Programme duration: Two years (four semesters) after obtaining a 3- year Bachelor's degree (e.g. B.A., B.Sc., B.Com.etc.). |
| Level 6.5 | Master's degree. (e.g. M.A., M.Com., M.Sc., etc.) Programme duration: One year (two semesters) after obtaining a 4 -year Bachelor's degree (Honours/ Honours with Research) (e.g. B.A., B.Sc., B.Com. etc.). |
| Level 7 | Master's degree.(e.g. M.E./M.Tech. etc.) Programme duration: Two years (four semesters) after obtaining a 4-year Bachelor's degree. (e.g. B.E./B.Tech. etc.) |
| Level 8 | Doctoral Degree |

3.  **NHEQF level descriptors**
Each NHEQF level is structured based on the defined learning outcomes which lead to the expected graduate attributes/profile. The level descriptors reflect the expected outcomes of learning that should be achieved and demonstrated by graduates of a specific programme of study leading to a qualification at a specific NHEQF level.

Click Learning outcomes descriptors for qualification for all levels on the NHEQF

4.  **Course Outcomes**
    The number of Course Outcomes is recommended to be 4-5 for courses that do not contain practical component and 6 for those courses with a practical component. Flexibility can be sought by the post-graduate courses in this regard.

5.  **Theory/lab Plan**
    The following are the guidelines to be followed while creating plans
- Each session may be planned for a duration of 45/50mins (irrespective of the double hour or single hour scheduled in timetable).
- Every session must incorporate at least one active learning method which may or may not be part of the assessments.
- Put BoS Approved Syllabus in the topics. Deviations (if any) from BoS approved syllabus must be brought to the notice of BoS chairman & Dean Academics, After approval, revised handout should be submitted.
- The Topics elaborated in the Theory/Lab plan must match those in the course execution plan.

24CSE0210/ Object Oriented Software Engineering

6. **Teaching Learning Methods**

The following are some of the Teaching & Learning methods that can be incorporated in session wise teaching learning plan.

- **Teacher-centered Learning Methods:**
    i. Lecture
    ii. Discussion
    iii. Demonstration method using a simulation or a tool
    iv. Reviewing
    v. Questioning

- **Learner-centered teaching & Learning methods:**
    i. **Active learning**, in which students solve problems, answer questions, formulate questions of their own, discuss, explain, debate, or brainstorm during class;
    ii. **Cooperative learning**, in which students work in teams on problems and projects under conditions that assure both positive interdependence and individual accountability; and
    iii. **Inductive teaching and learning**, in which students are first presented with challenges (questions or problems) and learn the course material in the context of addressing the challenges.
    iv. **Inductive methods** include inquiry-based learning, case based instruction, problem-based learning, project-based learning, discovery learning, and just-in-time teaching. It is important to integrate authentic, reflective and collaborative learning experiences when designing for student-centered learning.

7. **Active Learning Methods**

The following are some of the Active Learning Methods that can be incorporated in session wise teaching learning plan.

- One Minute Paper
- Group Discussion
- Student-Created PPT, Charts, Matrices, Flowcharts, Models
- The Fish Bowl
- Debate
- Video Synthesis
- Quiz/Test Questions
- Brain Storming Sessions
- Case Study
- Shadowing
- Leading Question
- Puzzle, Enigma, Contradiction
- Statement-Opinion-Summary
- Think / Pair / Share
- Peer Review
- Just in Time Teaching
- Statement-Opinion-Summary
- Peer Survey
- Focused Listing
- Role-Playing
- Student Field Work with Reflection
- Infusing Humor into Class Sessions
- Inviting Effective Guest Speakers

8. **Remedial Classes**

After every Sessional Test, identify weak learners, provide supplement course handout. Student list and Impact Observed report should be submitted to Dean through proper channel.

9. **Self Learning**

Plan 10% of topics in self-learning mode with discussions, ALM's and Assessment happing in the class.

10. **Content Beyond Syllabus**

Plan Advanced Topics, Experiments, Additional Reading, Research papers in self-learning mode with ALM's and Assessment happing in the regular class or lab. Usually caters advanced learners. Identify Advanced learners. For Extra classes, schedule should be notified accordingly.

11. **Assessment Type**

1. Assessment broadly can be classified into the following types:

a. **Diagnostic assessments**: Diagnostic assessments are intended to help teachers identify what students know and can do in different domains to support their students' learning. These help teachers determine strengths of students in various areas to better address their specific needs.

b. **Formative assessments**: Formative assessment refers to a wide variety of methods that teachers use to conduct in-process evaluations of student comprehension, learning needs, and academic progress during a lesson, unit, or a course. Formative assessments help teachers identify concepts that students are struggling to understand, skills they are having difficulty acquiring, or learning standards they have not yet achieved so that adjustments can be made to lessons, instructional techniques, and academic support.

c. **Summative assessments**: Summative assessment is an assessment administered at the end of an instructional unit in a course. These assessments are intended to evaluate student learning by comparing performance to a standard or benchmark.

d. **Ipsative assessments**: Ipsative assessment involves comparisons between past and current work to identify a learner's growth over time, rather than progress toward an external set of criteria. Therefore, Ipsative assessment is an internal or self-referenced assessment.

e. **Norm-referenced assessments**: Norm-referenced tests report whether test takers performed better or worse than a hypothetical average student, which is determined by comparing scores against the performance results of a statistically selected group of test takers, typically of the same age

24CSE0210/ Object Oriented Software Engineering

or grade level, who have already taken the exam.

f. **Criterion-referenced assessments**: Criterion-Reference tests measure the performance of test takers against the criteria covered in the curriculum.

g. **Peer-to-Peer randomised Assessments**: Peers will be able to provide assessment in this case

h. **Industry Validation of Effectiveness**: In the Vocation Education, Industry validation of effectiveness of training is particularly important.

i. **Self-assessments**: To evaluate how much the learner has grasped by self-learning.

2. Other Assessment Methods: Conducting an assessment takes time, thought, attention, planning, and often collaboration. Each assessment tool, whether a short survey or detailed rubric, will be useful only insofar as it both addresses the outcomes well and is feasible to use.

a**. Rubrics**: For assessing qualitative student work such as essays, projects, reports, or presentations. Rubrics serve well to clearly denote the specific expectations for an assignment, for collecting data for assessment of student learning outcomes. and for student performance. Rubrics can be used for grading, for providing feedback to students, and for informing and encouraging students to think about their own learning.

b. **Portfolios and E-Portfolio**: Portfolios can provide a window into the process of student learning across a semester-long project that can be assessed (usually by using a rubric).

c. **Curriculum Mapping**: A good curriculum map can serve to focus assessment, and the improvements that follow, where it will be most useful, informative, or effective.

d. **Structured Interviews**: While time-consuming, structured interviews are useful when specific questions need to be asked. It also leaves room for unplanned topics or ideas to emerge.

e. **Student Experience Surveys**: Student experience in research universities (SERU), including administration of on-line census SERU Undergraduate and Graduate Surveys, can yield important information about student perceptions and experiences.

12. **Evaluation Component & Types**

As per LMs we need to figure it out whether it is component 1, 2 or 3. In Types of Evaluation Component, we need to specify what type of evaluation we are performing like Continuous Evaluation or Sessional Test or End Term Examination.

13. **No. of Assessments and Weightage of Components**

Department will give guideline for number of assessments, mandatory or optional and weightage.