SRM VALLIAMMAI ENGINEERING COLLEGE

SRM Nagar, Kattankulathur – 603 203

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

QUESTION BANK

II SEMESTER

MC4265-OBJECT ORIENTED SOFTWARE ENGINEERING

Regulation – 2024

Academic Year 2024 – 2025 EVEN

Prepared by

Dr. R. Thenmozhi – Associate Professor / AI&DS

Dr.D.Sridevi-Associate Professor / IT

SRM VALLIAMMAI ENGNIEERING COLLEGE

SRM Nagar, Kattankulathur – 603203.

DEPARTMENT OF COMPUTER APPLICATIONS

QUESTION BANK

SUBJECT : MC4265- OBJECT ORIENTED SOFTWARE ENGINEERING

SEM / YEAR : II/ I

UNIT I - SOFTWARE DEVELOPMENT AND PROCESS MODELS

Introduction to Software Development – Challenges – An Engineering Perspective – Object Orientation – Software Development Process – Iterative Development Process – Process Models – Life Cycle Models – Unified Process – Iterative and Incremental – Agile Processes.

PART-A (2 - MARKS)

Q. No	QUESTIONS	Competence	BT Level	CO'S
1.	Write the IEEE definition of software engineering.	BTL1	Remembering	CO1
2.	Demonstrate your understanding of umbrella activities of a software process.	BTL2	Understanding	CO1
3.	List the goals of software engineering?	BTL1	Remembering	CO1
4.	List the characteristics of software contrasting it with characteristics of hardware.	BTL1	Remembering	CO1
5.	Express How do we create a process that can manage unpredictability?	BTL2	Understanding	CO1
6.	State the significance of life cycle models in software development?	BTL1	Remembering	CO1
7.	List two characteristics of software as a product.	BTL1	Remembering	CO1
8.	List two deficiencies in waterfall model. Which process model do you suggest to overcome each deficiency?	BTL1	Remembering	CO1
9.	Compare perspective and specialized process model.	BTL2	Understanding	CO1
10.	Outline the drawbacks of spiral model.	BTL2	Understanding	CO1
11.	Summarize the generic process framework that is applicable to any software project / relationship between work product, task, activity and system.	BTL2	Understanding	CO1
12.	What are the primary objectives of software life cycle models?	BTL2	Understanding	CO1
13.	Summarize the pros and cons of iterative software development model.	BTL2	Understanding	CO1
14.	Show what led to the transition from product oriented development to process oriented development.	BTL1	Remembering	CO1
15.	What are the potential advantages of adhering to life cycle models for software?	BTL2	Understanding	CO1
16.	Compare and contrast the relative advantages of object oriented and function oriented approaches to software design.	BTL2	Understanding	CO1
17.	What is the Unified Process?	BTL1	Remembering	CO1
18.	Describe the role of object orientation in software development.	BTL2	Understanding	CO1
19.	What are the advantages of iterative and incremental development?	BTL1	Remembering	CO1
20.	Define software maintenance.	BTL1	Remembering	CO1
21.	What are the benefits of using the Unified Process in software development?	BTL2	Understanding	CO1
22.	How do Agile processes promote customer collaboration?	BTL2	Understanding	CO1
23.	Define agile process .Give any two agile principles.	BTL1	Remembering	CO1
24.	Identify the human factors considered for an agile software development.	BTL2	Understanding	CO1

PART-B (16- MARKS)					
1.	Examine software life cycle. Classify all life cycle models and explain spiral model with a neat diagram.	(16)	BTL3	Applying	CO1
2.	(i) Explain one scenario where a) RAD model would be applicable and not the waterfall model. b) Waterfall model is preferable compare to all other models. (ii) What are the pros and cons of using mathematical approach for software development?	(8) (8)	BTL4	Analyzing	CO1
3.	(i) Describe about agile modeling in detail. (ii) Explain the component based software development model with a neat sketch	(8) (8)	BTL4	Analyzing	CO1
4.	(i) Write short notes on aspect oriented software development. (ii) Explain in detail about personal process models and team process models.	(8) (8)	BTL5	Evaluating	CO1
5.	Assume that you are the technical manager of a software development organization. A Client approached you for a software solution the problems stated by the client have uncertainties which lead to loss if it not planned and solved which software development model you will suggest for this project – justify. Explain that model With its pros and cons and neat sketch.	(16)	BTL6	Creating	CO1
6.	(i) Compare the life cycle models based on their distinguishing factors, strengths and weaknesses. (ii) Discuss the prototyping model .what is the effect of designing prototype on the overall cost of the software project?	(8) (8)	BTL4	Analyzing	CO1
7.	Explain in detail about iterative and waterfall model and also write short notes on concurrent models.	(16)	BTL4	Analyzing	CO1
8.	(i) Discuss in detail about drawback of life cycle model. (ii) What is the significance of the spiral model when compared with other model?	(8) (8)	BTL4	Analyzing	CO1
9.	Asses the Extreme Programming process and What are some of the issues that lead to an XP debate?	(16)	BTL5	Evaluating	CO1
10.	(i) Illustrate about agility and cost of change. (ii) What key traits must exist among the people on an effective software team?	(8) (8)	BTL3	Applying	CO1
11.	Examine about agility in the context of software engineering work? Organize the principles of agile software development.	(16)	BTL4	Analyzing	CO1
12.	(i) Compose your view about agile software development. (ii) Generalize your view about extreme programming.	(8) (8)	BTL4	Analyzing	CO1
13.	Compare the following life cycle models based on their distinguishing factors, strengths and weakness-waterfall model, RAD model, Spiral Model and Formal Methods Model.(Present in the form of table only-use diagrams wherever necessary).	(16)	BTL4	Analyzing	CO1
14.	Develop the umbrella activities which support software development process and discuss about their necessity in maintaining the quality in both software process and product that is being developed for railway reservation system.	(16)	BTL6	Creating	CO1
15.	Which process model is best suited for risk management? Discuss in detail with an example. Give the advantages and disadvantages of the model.	(16)	BTL5	Evaluating	CO1
16.	Analyze the concept of Agility. List the principles of agility and illustrate the Process in detail.	(16)	BTL4	Analyzing	CO1
17.	Demonstrate the XP concept of refactoring and pair programming	(16)	BTL5	Evaluating	CO1

UNIT II- MODELING OO SYSTEMS

Object Oriented Analysis (OOA / Coad-Yourdon), Object Oriented Design (OOD/Booch), Hierarchical Object Oriented Design (HOOD), Object Modeling Technique (OMT) – Requirement Elicitation – Use Cases – SRS Document – OOA - Identification of Classes and Relationships, Identifying State and Behavior – OOD - Interaction Diagrams – Sequence Diagram – Collaboration Diagrams - Unified Modeling Language and Tools.

PART-A (2 - MARKS)

Q.No.	QUESTIONS	BT Level	Competence	CO'S
1.	What is Object-Oriented Analysis (OOA)?	BTL1	Understanding	CO2
2.	How does the hierarchical approach in HOOD aid in object-oriented design?	BTL2	Remembering	CO2
3.	What are the phases in Booch's Object-Oriented Design?	BTL1	Understanding	CO2
4.	What are the different types of relationships in OOD?	BTL1	Understanding	CO2
5.	What are the main steps in the Coad-Yourdon OOA methodology?	BTL1	Understanding	CO2
6.	Define "generalization" in the Coad-Yourdon approach.	BTL1	Understanding	CO2
7.	When to use usecase diagram?	BTL1	Understanding	CO2
8.	How requirements are classified? List them with an example for each.	BTL2	Remembering	CO2
9.	Differentiate between normal and exciting requirement.	BTL2	Remembering	CO2
10.	Point out the problems faced when user requirements are written in natural language.	BTL2	Remembering	CO2
11.	What is the key feature of HOOD?	BTL1	Understanding	CO2
12.	What types of relationships are represented in HOOD?	BTL1	Understanding	CO2
13.	What are objects in OMT?	BTL1	Understanding	CO2
14.	What are the main diagrams used in OMT?	BTL1	Understanding	CO2
15.	List the characteristics of a good system requirements specification(SRS)	BTL1	Understanding	CO2
16.	Express the different types of check list that should be carried out for requirement validation process.	BTL2	Remembering	CO2
17.	Explain how to manage changing requirements during the requirements elicitation process?	BTL2	Remembering	CO2
18.	Define Petri Net and list types of traceability in a software process.	BTL1	Understanding	CO2
19.	Explain, how the requirements are validated?	BTL2	Remembering	CO2
20.	Define activity diagram?	BTL1	Understanding	CO2
21.	What is the use of system sequence diagram?	BTL1	Understanding	CO2
22.	Draw and explain a simple semantic data model for a library Management system	BTL2	Remembering	CO2
23.	What are all the various types of diagram that can be drawn in UML?	BTL2	Remembering	CO2
24.	List out any two Unified Modeling Language tools and its features	BTL2	Remembering	CO2

PART-B (16- MARKS)

1.	(i) Apply the Coad-Yourdon OOA method to analyze an online shopping system. (ii) Identify objects, attributes, and behaviors of online shopping system.	(8) (8)	BTL3	Applying	CO2
2.	Compare and contrast the Coad-Yourdon OOA method with Booch's object-oriented analysis methodology.	(16)	BTL4	Analyzing	CO2
3.	Evaluate the strengths and weaknesses of Booch's Object-Oriented Design methodology in large-scale software development.	(16)	BTL5	Evaluating	CO2

4.	Analyze the advantages of HOOD over other object-oriented design methodologies.	(16)	BTL4	Analyzing	CO2
5.	Using OMT, design a student information system by representing classes, relationships, and behaviors.	(16)	BTL3	Applying	CO2
6.	(i) Analyze the three models in OMT (ii)Explain object, dynamic, and functional models interdependencies.	(8) (8)	BTL4	Analyzing	CO2
7.	Explain in detail about the functional and non-functional requirements.	(16)	BTL4	Analyzing	CO2
8.	Show the template of IEEE standard software requirement document.	(16)	BTL3	Applying	CO2
9.	Explain the software requirement engineering process with neat diagram.	(16)	BTL3	Applying	CO2
10.	**Assess** on software requirement specification for banking system.	(16)	BTL5	Evaluating	CO2
11.	(i)What are system sequence diagram? (ii)What is the relationship between SSDs and Use cases? Explain with an example.	(04) (12)	BTL4	Analyzing	CO2
12.	Draw the following UML diagrams for online course reservation system. (i) Sequence diagram (ii) collaboration diagram	(8) (8)	BTL6	Creating	CO2
13.	Develop the process of ordering a pizza over the phone. Draw the use case diagram and also sketch the activity diagram representing each step of the process, from the moment you pick up the phone to the point where you start eating the pizza. Include activities that others need to perform. Add exception handling to the activity diagram you developed. Consider at least two exceptions (e.g. Delivery person wrote down wrong address, deliver person brings wrong pizza).	(16)	BTL6	Creating	CO2
14.	Compare and contrast association, aggregation, composition relationships with suitable examples.	(16)	BTL4	Analyzing	CO2
15.	(i) Examine the role of UML tools in software modeling and development. (ii) Critically assess the impact of UML tools on improving software documentation and maintainability.	(8) (8)	BTL4	Analyzing	CO2
16.	(i) Examine the role of interaction diagrams in understanding system behavior and object communication. (ii) Critically assess the limitations of interaction diagrams in large-scale system design.	(8) (8)	BTL4	Analyzing	CO2
17.	(i)Develop an online railway reservation system, which allows the user to select route, book/cancel tickets using net banking/credit /debit cards. The site also maintains the history of the passengers. (ii) Draw the use case scenario and model the specification for the above online railway reservation system.	(8) (8)	BTL6	Creating	CO2

UNIT III- DESIGN PATTERNS

Design Principles – Design Patterns – GRASP – GoF – Dynamic Object Modeling – Static ObjectModeling.

PART-A (2 - MARKS)		BT Level	Competence	CO'S
1.	What does the principle of "Separation of Concerns" mean in software design?	BTL1	Understanding	CO3
2.	How does the principle of "Consistency" support user experience in software systems?	BTL1	Understanding	CO3
3.	'A system must be loosely coupled and highly cohesive'. Compare	BTL2	Remembering	CO3
4.	What does GRASP stand for in the context of software engineering?	BTL1	Understanding	CO3
5.	What is the purpose of the Singleton Pattern in software design	BTL2	Remembering	CO3
6.	What does the Factory Method Pattern accomplish in object-oriented design?	BTL2	Remembering	CO3
7.	How does the Decorator Pattern differ from inheritance in extending an object's behavior?	BTL2	Remembering	CO3
8.	Why is the Strategy Pattern valuable in software design according to Pressman?	BTL2	Remembering	CO3
9.	What is the primary focus of the Controller pattern in GRASP?	BTL1	Understanding	CO3
10.	What does the Low Coupling principle in GRASP emphasize?	BTL1	Understanding	CO3
11.	Define the Creator pattern in GRASP.	BTL1	Understanding	CO3
12.	What are the advantages of using the Decorator Pattern as discussed by the GoF?	BTL2	Remembering	CO3
13.	How does the Controller pattern help in the separation of concerns in object-oriented design?	BTL2	Remembering	CO3
14.	Explain how the Information Expert pattern leads to better design in object-oriented systems.	BTL2	Remembering	CO3
15.	What is the purpose of the Abstract Factory Pattern as described by the GoF?	BTL2	Remembering	CO3
16.	Define the Strategy Pattern from the GoF design patterns.	BTL2	Remembering	CO3
17.	Describe how the Observer Pattern improves decoupling between components in software design.	BTL2	Remembering	CO3
18.	Explain how the Information Expert pattern leads to better design in object-oriented systems.	BTL2	Remembering	CO3
19.	What **is** Dynamic Object Modeling in software engineering? Name any two common diagrams used in Dynamic Object Modeling.	BTL1	Understanding	CO3
20.	What is the role of Events and Transitions in a State Diagram?	BTL1	Understanding	CO3
21.	How **do** Activity Diagrams complement State Diagrams in Dynamic Object Modeling?	BTL2	Remembering	CO3
22.	Explain the significance of Aggregation and Composition in a Class Diagram.	BTL2	Remembering	CO3
23.	Why is Generalization (Inheritance) important in Static Object Modeling?	BTL2	Remembering	CO3
24.	What is the purpose of a State Diagram in Dynamic Object Modeling?	BTL1	Understanding	CO3

PART-B (16 - MARKS)					
1.	(i) How can the Model-View-Controller (MVC) architecture be applied in web application development? (ii)Provide an example and discuss its advantages.	(8) (8)	BTL3	Applying	CO3
2.	Compare and contrast the Monolithic Architecture vs. Micro services Architecture in terms of the following: (i)Design principles (ii)Scalability (iii)Maintainability.	(6) (5) (5)	BTL4	Analyzing	CO3
3.	(i).Assess the role of Agile Design Principles in modern software engineering. (ii) Are they universally applicable to all projects?	(10) (6)	BTL5	Evaluating	CO3

4.	Discuss the Gang of Four (GoF) design patterns with suitable examples.	(16)	BTL3	Applying	CO3
5.	Compare and contrast creational, structural, and behavioral patterns with examples.	(16)	BTL4	Analyzing	CO3
6.	(i) Describe the Factory Method Pattern and its role in object creation. (ii) Illustrate Factory Method Pattern with an example.	(8) (8)	BTL5	Evaluating	CO3
7.	Develop a Decorator Pattern example to dynamically add functionalities to a graphical user interface (GUI) component.	(16)	BTL3	Applying	CO3
8.	Compare and contrast the Adapter Pattern and Proxy Pattern with real-world scenarios.	(16)	BTL4	Analyzing	CO3
9.	Create a Static Object Model for an online food delivery platform integrating customer, restaurant, and order entities.	(16)	BTL6	Creating	CO3
10.	(i) Compare and contrast High Cohesion and Low Coupling (ii) Explain High Cohesion and Low Coupling with real-world software examples.	(8) (8)	BTL4	Analyzing	CO3
11.	Critically evaluate the effectiveness of GRASP principles in modern software development**.**	(16)	BTL5	Evaluating	CO3
12.	Design an automated traffic management system using GRASP principles for responsibility assignment.	(16)	BTL6	Creating	CO3
13.	Develop a Class Diagram with Inheritance and Association for an online shopping application**.**	(16)	BTL4	Analyzing	CO3
14.	Compare and contrast Aggregation and Composition relationships in Static Object Modeling**.**	(16)	BTL4	Analyzing	CO3
15.	Implement a Sequence Diagram for **a** hotel booking system to illustrate object interactions.	(16)	BTL3	Applying	CO3
16.	Apply State Diagrams to model the behavior of an ATM machine during a transaction.	(16)	BTL3	Applying	CO3
17.	Analyze the role of Sequence Diagrams in UML-based software design with a real-world example.	(16)	BTL4	Analyzing	CO3

UNIT IV- SYSTEM TESTING

Software testing: Software Verification Techniques – Object Oriented Checklist :- Functional Testing– Structural Testing – Class Testing – Mutation Testing – Levels of Testing – Static and Dynamic Testing Tools - Software Maintenance – Categories – Challenges of Software Maintenance –Maintenance of Object Oriented Software – Regression Testing

PART-A (2 -MARKS)		BT Level	Competence	CO'S
1.	What is the difference between black box testing and white box testing?	BTL2	Understanding	CO4
2.	What methods are used for breaking very long expression and statements?	BTL1	Remembering	CO4
3.	What is the need for regression testing and system testing?	BTL1	Remembering	CO4
4.	List the levels of testing.	BTL1	Remembering	CO4
5.	How do you measure cyclomatic complexity?	BTL2	Understanding	CO4
6.	What is a test case?	BTL1	Remembering	CO4
7.	Describe about software maintenance problem.	BTL2	Understanding	CO4
8.	Define boundary value analysis.	BTL1	Remembering	CO4
9.	How can refactoring be made more effective?	BTL2	Understanding	CO4
10.	List the different levels of testing in software development.	BTL1	Remembering	CO4

11.	How does Mutation Testing help improve the effectiveness of a test suite?	BTL2	Understanding	CO4
12.	What are Static Testing Tools used for in software testing?	BTL1	Remembering	CO4
13.	Distinguish between alpha and beta testing.	BTL2	Understanding	CO4
14.	List two testing strategies that address verification. Which types of testing address validation?	BTL1	Remembering	CO4
15.	How do Dynamic Testing Tools help in assessing the runtime behavior of a system?	BTL2	Understanding	CO4
16.	Differentiate verification and validation. Which type of testing address verification?	BTL2	Understanding	CO4
17.	What happen if the software fails after it has passed from acceptance testing?	BTL2	Understanding	CO4
18.	List the different categories of Software Maintenance.	BTL1	Remembering	CO4
19.	What is Software Maintenance in the context of software development?	BTL2	Understanding	CO4
20.	Who Should perform the validation test, software developer or the software users?	BTL1	Remembering	CO4
21.	How do legacy systems pose a challenge in software maintenance?	BTL2	Understanding	CO4
22.	Why is poor documentation a major challenge in software maintenance?	BTL2	Understanding	CO4
23.	List the different types of changes that can occur in object-oriented software maintenance.	BTL1	Remembering	CO4
24.	Define Regression Testing in software maintenance.	BTL1	Remembering	CO4

PART-B (16- MARKS)

1.	Compare the following i. Unit testing &Regression testing ii. Validation testing & Acceptance testing	(8) (8)	BTL4	Analyzing	CO4
2.	(i) What is Boundary value analysis? (ii) Explain the technique specifying rules and is usage with the help of an example.	(4) (12)	BTL4	Analyzing	CO4
3.	(i)What is Equivalence class partitioning? (ii)List rules used to define valid and invalid Equivalence class and Demonstrate the Equivalence class partitioning technique using example.	(4) (12)	BTL3	Applying	CO4
4.	Elaborate path testing with an example.	(16)	BTL3	Applying	CO4
5.	Discuss the various Black box and white Box testing techniques. Use suitable example for your explanation**.**	(16)	BTL3	Applying	CO4
6.	Compare the various Integration strategies & debugging followed in software development.	(16)	BTL4	Analyzing	CO4
7.	(i) Explain software implementation techniques .What is the percentage in total cost of the project? How do you expedite the implementation stage	(8)	BTL5	Evaluating	CO4
	(ii) What is meant by control flow testing? Is it always falling with dataflow in case of software? Justify?	(8)	BTL5	Evaluating	CO4
8.	(i) What is black box testing? (ii)Explain the different types of black box testing strategies with example?	(8) (8)	BTL6	Creating	CO4
9.	(i) Explain the categories of debugging approaches.	(8)	BTL4	Analyzing	CO4
	(ii) Why is testing important? Relate the path testing procedure in detail with sample code.	(8)	BTL4	Analyzing	CO4
10.	Compare and contrast corrective and preventive maintenance, explaining their importance in software lifecycle.	(16)	BTL4	Analyzing	CO4
11.	Evaluate the effectiveness of automated testing tools in reducing manual testing efforts**.**	(16)	BTL5	Evaluating	CO4

12.	Construct a preventive maintenance strategy for reducing software failures in a cloud-based application.	(16)	BTL3	Applying	CO4
13.	(i) Enumerate the various types of software test. Which type of testing is suitable for boundary condition? Justify. (ii) How do you relate software testing results with reliability of the product? Explain.	(8) (8)	BTL6	Creating	CO4
14.	Describe the type's basic path testing given. (i)Flow graph notation. (ii) Independent program paths.	 (8) (8)	BTL4	Analyzing	CO4
15.	Critically assess the challenges of maintaining legacy software systems in modern enterprises.	(16)	BTL5	Evaluating	CO4
16.	Analyze the importance of Regression Testing in risk management for software releases.	(16)	BTL4	Analyzing	CO4
17.	Design a software maintenance plan for a large-scale healthcare management system.	(16)	BTL6	Creating	CO4

UNIT V- SOFTWARE QUALITY AND METRICS

Need of Object Oriented Software Estimation – Lorenz and Kidd Estimation – Use Case Points Method – Class Point Method – Object Oriented Function Point – Risk Management – SoftwareQuality Models – Analyzing the Metric Data – Metrics for Measuring Size and Structure – Measuring Software Quality - Object Oriented Metrics.

PART-A (2 -MARKS)		BT Level	Competence	CO'S
1.	Name two object-oriented metrics used in software estimation.	BTL1	Remembering	CO5
2.	Define Function Point Analysis in software estimation.	BTL2	Understanding	CO5
3.	How does object-oriented estimation differ from traditional estimation techniques?	BTL2	Understanding	CO5
4.	What is the primary purpose of Lorenz and Kidd estimation?	BTL1	Remembering	CO5
5.	What are the three main complexity categories used in Lorenz and Kidd estimation?	BTL1	Remembering	CO5
6.	Explain the significance of object-oriented metrics in Lorenz and Kidd estimation.	BTL2	Understanding	CO5
7.	Define Unadjusted Use Case Points (UUCP).	BTL1	Remembering	CO5
8.	Why are Environmental Factors considered in UCP estimation?	BTL2	Understanding	CO5
9.	How does the UCP method differ from Function Point Analysis?	BTL2	Understanding	CO5
10.	How is class complexity classified in the Class Point Method?	BTL1	Remembering	CO5
11.	What are the advantages of using the Class Point Method in software estimation?	BTL2	Understanding	CO5
12.	Explain how the complexity of a class is determined in the Class Point Method.	BTL2	Understanding	CO5
13.	What is the primary purpose of Object-Oriented Function Points?	BTL1	Remembering	CO5
14.	What is the role of object interactions in OOFP estimation?	BTL1	Remembering	CO5
15.	What is the significance of encapsulation in OOFP calculation?	BTL2	Understanding	CO5
16.	How can the OOFP method improve software project planning and estimation?	BTL2	Understanding	CO5
17.	Define risk identification.	BTL1	Remembering	CO5
18.	Name two risk mitigation strategies.	BTL1	Remembering	CO5
19.	Explain the difference between qualitative and quantitative risk analysis.	BTL2	Understanding	CO5
20.	What is the role of stakeholders in risk management?	BTL1	Remembering	CO5
21.	List the key phases of risk management.	BTL1	Remembering	CO5
22.	Why is software quality important in software development?	BTL2	Understanding	CO5
23.	What is the role of software quality attributes in project success?	BTL1	Remembering	CO5
24.	What is a software quality model?	BTL1	Remembering	CO5

PART-B(16 MARKS)					
1.	(i)Compare and contrast Lorenz and Kidd Estimation **with** COCOMO (ii) Highlighting their applicability in different project sizes.	(8) (8)	BTL4	Analyzing	CO5

2.	Differentiate between size-based estimation of Lorenz and Kidd and expert judgment-based estimation techniques of Delphi method.	(16)	BTL4	Analyzing	CO5
3.	How the cost of a software is estimated using (i) Function Point metric Model & COCOMO(by three Methods.) (ii)What is the contribution of technology complexity factor in function point model?	(10) (6)	BTL3	Applying	CO5
4.	Justify the need for Object-Oriented Software Estimation (OOSE) in large-scale projects with multiple interacting components	(16)	BTL5	Evaluating	CO5
5.	Differentiate between simple, medium, and complex classes in the context of Class Point Estimation.	(16)	BTL4	Analyzing	CO5
6.	(i)Demonstrate the basic principles of software project scheduling. (ii)Apply the relationship between people and effort with diagram.	(8) (8)	BTL3	Applying	CO5
7.	(i) Point out the challenges of risk management. (ii)How to track the schedule for the project? Explain in detail.	(8) (8)	BTL4	Analyzing	CO5
8.	Analyze the role of actors and use case complexity in the Use Case Points method and how they influence effort estimation.	(16)	BTL4	Analyzing	CO5
9.	State the need for Risk Management &explain the activities under risk management.	(16)	BTL4	Analyzing	CO5
10.	Demonstrate the following (i) Project scheduling. (ii) Project Time Line chart & Task network.	(8) (8)	BTL3	Applying	CO5
11.	List the features of LOC and FP based estimation models and Compare the two models and list the advantages of one over other.	(16)	BTL4	Analyzing	CO5
12.	Analyze how coupling and cohesion in class design affect class point estimation and software development effort.	(16)	BTL4	Analyzing	CO5
13	Assess the accuracy and limitations of OOFP in estimating the development cost of highly complex object-oriented applications**.**	(16)	BTL5	Evaluating	CO5
14.	(i) Explain the key quality factors in Boehm's model, such as product revision, product operation, and product transition**.** (ii) Assess how this model helps project managers in making quality-related decisions during software development.	(16)	BTL5	Evaluating	CO5
15.	(i) Examine risk management in a software development life cycle. (ii) Analyse the concept of RMMM.	(8) (8)	BTL4	Analyzing	CO5
16.	(i) Explain the role of metrics in evaluating the effectiveness of software quality models**.** (ii) How can customer satisfaction metrics be linked to the software quality model during product release?	(8) (8)	BTL3	Applying	CO5
17.	(i) Explain in detail about risk identification. (ii)Analyze on the concept of risk Projection.	(8) (8)	BTL4	Analyzing	CO5

\`\`\`