23CMU419N

PSG COLLEGE OF ARTS & SCIENCE
(AUTONOMOUS)

BSc DEGREE EXAMINATION MAY 2025
(Fourth Semester)

Branch – **COMPUTER SCIENCE**

SOFTWARE ENGINEERING & TESTING

Time: Three Hours

Maximum: 75 Marks

SECTION-A (10 Marks)

Answer **ALL** questions

ALL questions carry **EQUAL** marks (10 × 1 = 10)

Module No.	Question No.	Question	K Level	CO
1	1	Identify the process model known for its sequential and linear approach. a) Agile Model b) Waterfall Model c) Iterative Model d) Spiral Model	K1	CO1
	2	Which UML diagram best represents the interactions between a system and its external actors? a) Class Diagram b) Sequence Diagram c) Use Case Diagram d) Component Diagram	K2	CO1
2	3	What type of testing identifies the internal logic and structure of the code? a) Black Box Testing b) White Box Testing c) User Acceptance Testing d) System Testing	K1	CO2
	4	Identify the option that is NOT categorized as a key component of an Object-Oriented Analysis (OOA) model. a) Use Case Diagrams b) Class Diagrams c) Sequence Diagrams d) Data Flow Diagrams	K2	CO2
3	5	Select the option that is NOT a core principle of Model-Driven Software Development (MDSD). a) Model-centricity b) Automation c) Sequential development d) Increased productivity	K1	CO3
	6	Determine the primary purpose of verification in the software development lifecycle. a) To ensure that the software meets the specified requirements. b) To ensure that the software is built correctly. c) To ensure that the software meets the user's needs and expectations. d) To ensure that the software is free from defects.	K2	CO3
4	7	State the type of performance testing that focuses on the system's behavior under extreme or peak load conditions a) Load Testing b) Stress Testing c) Volume Testing d) Soak Testing	K1	CO4
	8	Explain the type of acceptance testing that involves end-users evaluating the software in a real-world environment. a) Alpha Testing b) Beta Testing c) User Acceptance Testing d) Integration Testing	K2	CO4

Cont...

5	9	Select the item that is NOT typically considered a test deliverable. a) Test plan document b) Test summary report c) Source code of the software under test d) Test incident reports	K1	CO5
	10	Describe the importance of maintaining a traceability matrix between test cases and requirements. a) To ensure that all requirements have been adequately tested b) To track the progress of the testing effort c) To identify and prioritize critical defects d) To generate automated test scripts	K2	CO5

SECTION - B (35 Marks)

Answer **ALL** questions

ALL questions carry **EQUAL** Marks (5 × 7 = 35)

Module No.	Question No.	Question	K Level	CO
1	11.a.	Describe the core principles of Agile development and analyze how they address the challenges of traditional software development methodologies. Provide examples to support your answer.	K2	CO1
		(OR)		
	11.b.	Outline the process of eliciting requirements from stakeholders and analyze the challenges involved. Discuss how effective requirement elicitation contributes to the success of a software project.		
2	12.a.	Evaluate the importance of a well-defined requirements model in the software development process. Justify the key characteristics of a good requirements model, including completeness, consistency, and testability.	K3	CO2
		(OR)		
	12.b.	Analyze the key concepts of Object-Oriented Analysis (OOA) and Object-Oriented Design (OOD). Justify how OOA and OOD models, such as use case diagrams and class diagrams, contribute to the creation of robust and maintainable software systems.		
3	13.a.	Evaluate the key goals and benefits of implementing a Software Process Improvement (SPI) initiative within an organization. Justify the challenges involved in implementing and sustaining an effective SPI program.	K3	CO3
		(OR)		
	13.b.	Analyze how Model-Driven Software Development (MDSD) can enhance the efficiency and effectiveness of the requirements engineering process. Justify the role of models in capturing and communicating requirements in an MDSD approach.		

Cont...

4	14.a.	Critique the principles and practices of testing in Agile development methodologies, such as Extreme Programming (XP). Evaluate the challenges of implementing effective testing practices within an Agile framework.	K4	CO4
		(OR)		
	14.b.	Assess how use case scenarios can guide the design and execution of integration tests. Evaluate the challenges of ensuring that all critical interactions between modules are adequately tested during the integration testing phase.		
5	15.a.	Evaluate the critical elements of a well-defined test plan. Analyze the importance of risk assessment and resource allocation in the test planning process.	K4	CO5
		(OR)		
	15.b.	Analyze the key characteristics of a well-designed test automation framework. Evaluate the essential components of an effective test automation framework, such as test data management, reporting, and integration with other tools.		

SECTION -C (30 Marks)

Answer **ANY THREE** questions

ALL questions carry **EQUAL** Marks (3 × 10 = 30)

Module No.	Question No.	Question	K Level	CO
1	16	Evaluate the concept of software architecture and its significance in the overall success of a software project. Analyze how architectural decisions impact key quality attributes such as performance, maintainability, scalability, and security.	K4	CO1
2	17	Analyze the role of various testing levels in ensuring software quality. Assess how these different levels of testing contribute to identifying and resolving defects throughout the software development lifecycle.	K4	CO2
3	18	Evaluate the suitability of different software development lifecycle models for the development of a large, complex software system with critical safety requirements.	K4	CO3
4	19	Formulate a strategy for effectively integrating white-box and black-box testing techniques to achieve comprehensive test coverage and improve the overall quality of a software system. Synthesize how different testing levels can be coordinated to create a cohesive testing strategy.	K5	CO4
5	20	Devise a strategy for incorporating continuous integration into an effective test automation framework. Synthesize how continuous integration practices can enhance the quality and speed of software delivery.	K5	CO5

Z-Z-Z END