



University of Colombo, Sri Lanka

University of Colombo School of Computing



**DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY
(EXTERNAL)**

Academic Year 2025— 1st Year Examination — Semester 2

IT2206 — Fundamentals of Software Engineering

Multiple Choice Question Paper
(2 Hours)

Important Instructions

- The duration of the paper is **2 Hours**.
- The medium of instructions and questions is English.
- This paper has **40 MCQ questions** on **9 pages**. Answer **all** questions.
- Each question will have **5 (five)** choices with **ONLY ONE** correct answer.
- This paper consists of 100 marks and all the questions will carry equal marks.
- Answers should be marked on the **special answer sheet** provided.
- Note that questions appear on both sides of the paper. If a page or part of a page is not printed, please inform the supervisor/invigilator immediately.
- Mark the correct choices on the question paper first and then transfer them to the given answer sheet which will be machine marked. **Please completely read and follow the instructions given on the other side of the answer sheet before you shade your correct choices.**
- Any electronic device capable of storing and retrieving text, including electronic dictionaries, smartwatches, and mobile phones, is not allowed.
- Calculators are **not** allowed.
- *All Rights Reserved.* This question paper can NOT be used without proper permission from the University of Colombo School of Computing.

- 1) Which of the following best describes software as a product?
- (a) A collection of programs and hardware devices.
 - (b) A collection of programs, configuration files and documentation.
 - (c) A collection of programs and databases only.
 - (d) A collection of databases and network only.
 - (e) A collection of processors and documentation.
- 2) Which of the following is considered as the main difference between generic and bespoke software?
- (a) Generic software is free while bespoke software is not.
 - (b) Generic software is developed for a broad market, while bespoke software is developed for a specific customer or organization.
 - (c) Generic software has no documentation while bespoke software will always produce documentation.
 - (d) Generic software can be easily modified while bespoke software cannot be modified.
 - (e) Generic software is always web-based while bespoke software is not.
- 3) Which of the following best explains why software engineering differs from many traditional engineering disciplines?
- (a) Software is always visible, and requirements are stable.
 - (b) Software requirements rarely change.
 - (c) Software is intangible, and requirements often change during development.
 - (d) Software is intangible and cannot be tested.
 - (e) Software is not reusable, and requirements rarely change.
- 4) Which of the following characteristics makes software systems often become more complex over time?
- (a) Software cannot be copied or reused.
 - (b) Software replication becomes difficult over time.
 - (c) Hardware dependence increases over time.
 - (d) Changes, fixes, and enhancements accumulate throughout the software life cycle.
 - (e) Software documentation is removed over time.
- 5) Which of the following activities defines what a software system should do?
- | | | |
|--------------------|-------------------|-----------------|
| (a) Validation | (b) Evolution | (c) Maintenance |
| (d) Implementation | (e) Specification | |

6) Which of the following set contains four fundamental software process activities?

- (a) Coding, Testing, Selling, Deployment
- (b) Specification, Design & Implementation, Validation, Evolution
- (c) Planning, Marketing, Coding, Maintenance
- (d) Analysis, Sales, Design, Repair
- (e) Requirements gathering, Accounting, Testing, Support

7) Which of the following best describes a software process model?

- (a) A programming language used to develop software systems.
- (b) A hardware architecture that supports software execution.
- (c) A technique used to test software applications.
- (d) A financial plan outlining project costs and expenditures.
- (e) An abstract representation of the activities and their workflow.

8) Which of the following activities checks whether software meets customer needs?

- | | | |
|-------------------|-----------------|----------------|
| (a) Specification | (b) Evolution | (c) Validation |
| (d) Procurement | (e) Refactoring | |

9) Which of the following is the primary purpose of a software process description?

- (a) To define the hardware infrastructure for software deployment.
- (b) To outline the development activities, their sequence, and their interrelationships.
- (c) To specify the syntax rules of the programming language used.
- (d) To document legal and licensing requirements.
- (e) To manage customer payments and invoices.

10) Which of the following is a key objective of agile software development?

- (a) To reduce customer involvement.
- (b) To avoid testing during development.
- (c) To eliminate all documentation.
- (d) To respond quickly to changing requirements.
- (e) To increase bureaucracy in the development process.

- 11) Which of the following factors is most commonly associated with software project failure?
- (a) Alignment of the system with user requirements.
 - (b) Effective user interface design.
 - (c) Unrealistic deadline pressure, poor usability, and weak quality control.
 - (d) Continuous stakeholder feedback throughout development.
 - (e) Incremental delivery of software features.
- 12) Which of the following best reflects a fundamental principle of Agile software development methodologies?
- (a) Emphasis on extensive upfront planning and design.
 - (b) Customer collaboration and responsiveness to changing requirements.
 - (c) Elimination of all project documentation.
 - (d) Strict adherence to fixed requirements throughout the project.
 - (e) Sequential execution of development phases.
- 13) Which of the following project characteristic would most strongly justify the adoption of an agile development methodology rather than a plan-driven methodology?
- (a) Requirements are well understood and unlikely to change.
 - (b) Regulatory requirements demand extensive upfront documentation.
 - (c) Frequent changes in requirements require rapid adaptation and stakeholder feedback.
 - (d) The project scope has been finalized and approved before development begins.
 - (e) The system has already been fully implemented.
- 14) Which of the following sets of activities is most closely associated with requirements engineering in a software project?
- (a) Coding, debugging, and deployment
 - (b) Requirements elicitation, analysis, specification, and validation
 - (c) Procurement, installation, and configuration of hardware
 - (d) Unit testing, integration testing, and system testing
 - (e) User training, deployment, and maintenance
- 15) The main purpose of user requirements is to communicate the required system services and constraints to:
- | | |
|-----------------------|--------------------------------|
| (a) Developers only | (b) Compilers |
| (c) Operating systems | (d) Customers and stakeholders |
| (e) Hardware devices | |

- 16) System requirements are typically documented as:
- (a) Informal notes and observations
 - (b) Detailed descriptions of system functions and constraints
 - (c) Marketing materials for stakeholders
 - (d) Test reports generated during validation
 - (e) User training manuals
- 17) A client provides high-level requirements without specifying technical details so that different vendors can propose their own solutions. This approach is mainly intended to:
- (a) ensure the system cannot be implemented until further clarification is added.
 - (b) allow different providers to propose alternative ways of achieving the outcome.
 - (c) reduce the need for discussions with users during development.
 - (d) avoid checking whether the requirements are correct or complete.
 - (e) guarantee that only one specific design approach is used.
- 18) Which of the following best distinguishes user requirements from system requirements?
- (a) User requirements are expressed in greater technical detail than system requirements.
 - (b) Both represent the same level of specification with no distinction.
 - (c) User requirements are always legal contract documents.
 - (d) System requirements never include performance or design constraints.
 - (e) System requirements describe the system in detailed terms including behavior and constraints.
- 19) During early system development, analysts create diagrams to represent system processes and interactions so that stakeholders and developers share a common understanding. This activity is known as:
- (a) System modeling
 - (b) Software coding
 - (c) Testing automation
 - (d) System deployment
 - (e) Hardware configuration
- 20) Which of the following is most commonly used as a notation for system modeling in software engineering?
- | | | |
|---------|----------|---------|
| (a) XML | (b) HTML | (c) SQL |
| (d) UML | (e) JSON | |

- 21) Which of the following is a key benefit of creating models of an existing system?
- (a) They make ongoing system maintenance unnecessary.
 - (b) They support analysis of current system behavior.
 - (c) They ensure all system defects are automatically removed.
 - (d) They replace the need for requirements engineering activities.
 - (e) They directly determine the financial success of a project.
- 22) During system design, analysts create diagrams showing how users and system components communicate with each other over time. This represents a:
- (a) Data storage perspective
 - (b) Business modeling perspective
 - (c) Financial analysis perspective
 - (d) Interaction perspective
 - (e) Resource allocation perspective
- 23) Which of the following perspective models the dynamic response of a system to events?
- (a) Structural
 - (b) External
 - (c) Behavioral
 - (d) Contextual
 - (e) Economic
- 24) Which of the following is a valid use of system models in Model-Driven Engineering (MDE)?
- (a) Eliminating the need for software testing.
 - (b) Replacing the requirements specification process.
 - (c) Removing stakeholder involvement from development.
 - (d) Generating complete or partial system implementations.
 - (e) Creating hardware prototypes.
- 25) Which of the following best describes the primary objective of architectural design in software engineering?
- (a) Defining the internal logic and algorithms of individual components.
 - (b) Specifying detailed user interface layouts and screen designs.
 - (c) Identifying and documenting system test cases and validation procedures.
 - (d) Selecting programming languages and development tools for implementation.
 - (e) Organizing the overall structure of the system and the relationships between its components.
- 26) Which of the following is the typical output of the architectural design process?
- (a) A program code
 - (b) An architectural model
 - (c) A user manual
 - (d) A test plan
 - (e) A user interface design

- 27) Which of the following best explains why architectural refactoring can be expensive in large software systems?
- (a) Changes to the architecture can affect many parts of the system.
 - (b) Architectural information is often not documented in detail.
 - (c) Architectural refactoring can only be performed during system maintenance.
 - (d) Refactoring mainly involves replacing hardware component.
 - (e) Architectural changes eliminate the need for further testing.
- 28) The term "architecture in the small" refers to the architecture of:
- (a) An enterprise-wide software ecosystem.
 - (b) A network of interconnected organizations.
 - (c) An individual software application or program.
 - (d) Business processes within an organization.
 - (e) Software project budgeting and resource allocation.
- 29) The term "architecture in the large" primarily concerns:
- (a) The organization of components within a single software application.
 - (b) The interaction and integration of multiple independent systems across an enterprise.
 - (c) The detailed design of classes, methods, and data structures.
 - (d) The deployment configuration of a standalone software product.
 - (e) The implementation of reusable modules within a software framework.
- 30) Implementation in software engineering is the process of:
- (a) Defining the functional and non-functional requirements of a system.
 - (b) Evaluating whether the system satisfies its specified requirements.
 - (c) Modifying a software system after deployment to address changes and defects.
 - (d) Organizing the overall structure and components of a software system.
 - (e) Realizing a software design as an executable program.
- 31) Which of the following is a major advantage of using Commercial Off-The-Shelf (COTS) software?
- (a) It can be developed to meet highly specialized organizational requirements.
 - (b) It eliminates the need for system integration and configuration activities.
 - (c) It can be acquired and deployed quickly and at lower cost than custom-built software.
 - (d) It provides complete control over the software source code and architecture.
 - (e) It guarantees that the software will satisfy all future business requirements.

- 32) Why are design models particularly important in the development of large software systems involving multiple teams?
- (a) They serve as a communication mechanism among stakeholders and developers.
 - (b) They provide a basis for automatic code generation without further development effort.
 - (c) They eliminate the need for system verification and validation activities.
 - (d) They reduce the need to document system requirements.
 - (e) They remove the need for future system maintenance and evolution.
- 33) Which of the following is the primary purpose of software testing?
- (a) To verify and identify defects in a software system before deployment.
 - (b) To define the functional and non-functional requirements of the system.
 - (c) To transform software design models into executable code.
 - (d) To improve the maintainability and adaptability of the software architecture.
 - (e) To produce technical and user documentation for system operation.
- 34) Which of the following best defines a software bug?
- (a) A change made to improve the functionality of a software system.
 - (b) A requirement that specifies how the software should operate.
 - (c) A software component that provides reusable functionality to other modules.
 - (d) An error or flaw in a software system that causes incorrect or unintended behavior.
 - (e) A test case designed to verify that the software meets its requirements.
- 35) Why is software testing particularly important for applications that serve a large number of users?
- (a) Different usage patterns may expose defects that are not evident during development.
 - (b) Testing ensures that all users will interact with the system in the same way.
 - (c) Testing eliminates the need for future maintenance and updates.
 - (d) Large user bases reduce the complexity of software systems.
 - (e) Software behavior remains unchanged regardless of the number of concurrent users.
- 36) Which of the following consequences is most closely associated with failures in safety-critical software systems?
- (a) Financial losses resulting from interrupted business operations.
 - (b) Injury to people or loss of human life.
 - (c) Reduced system performance and response times.
 - (d) Delays in software maintenance and upgrades.
 - (e) Increased operational and training costs.

- 37) Which of the following is **NOT** a benefit that can be achieved through effective software testing?
- (a) Increased software reliability and quality.
 - (b) Greater confidence among users and stakeholders.
 - (c) Reduced costs associated with defect correction and maintenance.
 - (d) Complete assurance that the software contains no defects.
 - (e) More consistent system behavior across different operating conditions.
- 38) Which of the following best explains why software systems must continually change after they are developed?
- (a) Requirements, operating environments, and technologies evolve over time.
 - (b) Once developed, software systems remain completely unchanged.
 - (c) Users no longer require updates after initial deployment.
 - (d) Testing activities eliminate the need for future modifications.
 - (e) Hardware platforms remain static and never influence software evolution.
- 39) Which of the following is the main reason organizations allocate significant resources to software evolution?
- (a) To reduce the overall value of existing software systems.
 - (b) To maintain business value of software assets.
 - (c) To eliminate the need for any future maintenance activities.
 - (d) To prevent users from requesting system changes.
 - (e) To avoid updating software after its initial release.
- 40) Which of the following activities typically consumes the largest portion of the software budget in large organizations?
- (a) Developing new software systems from the ground up.
 - (b) Purchasing office equipment and furniture.
 - (c) Conducting marketing and promotional activities.
 - (d) Maintaining, modifying, and evolving existing software systems.
 - (e) Installing physical networking infrastructure.
