

Class 11 Computer Science

Introduction to Software Development

Complete Solved Exercise (English Medium)

Multiple Choice Questions (MCQs)

1. What is the primary purpose of the Software Development Life Cycle (SDLC)?

- a) To design websites
- ✓ b) To deliver high-quality software within time and cost estimates
- c) To manage database systems
- d) To create hardware components

2. Which type of requirement specifies how the system should perform?

- a) Functional Requirements
- ✓ b) Non-Functional Requirements
- c) Technical Requirements
- d) Operational Requirements

3. In the context of SDLC, what is the role of a framework?

- a) To write code from scratch
- ✓ b) To provide a structured foundation with predefined components and architectures
- c) To manage hardware
- d) To perform manual testing

4. Which software development model involves working in short cycles or sprints?

- a) Waterfall Model
- b) Agile Methodology
- c) Lean Software Development
- ✓ d) Scrum

5. Which role is responsible for removing obstacles and facilitating Scrum practices?

- a) Product Owner
- ✓ b) Scrum Master
- c) Development Team
- d) Project Manager

6. Which of the following is not a benefit of DevOps?

- a) Improved collaboration
- b) Enhanced quality
- ✓ c) Increased project scope creep
- d) Faster time-to-market

7. What is a crucial aspect of comprehensive project planning?

- ✓ a) Understanding the project scope and tasks
- b) Deciding the project's colour scheme
- c) Hiring a large development team
- d) Ignoring potential risks

8. Which factor does NOT influence the cost estimation of a software project?

- a) Scope of the project
- b) Technology stack
- ✓ c) Number of meetings held
- d) Operational costs

9. What is the purpose of a contingency fund in cost estimation?

- ✓ a) To cover unexpected costs
- b) To pay for marketing expenses
- c) To hire additional developers
- d) To purchase new hardware

10. Which of the following is a purpose of Use Case Diagrams?

- a) To document the system's architecture
- ✓ b) To identify and document the system's functional requirements
- c) To illustrate the database schema
- d) To define the system's user interface design

Short Questions

1. Differentiate between functional and non-functional requirements.

Functional requirements describe specific behavior or functions of the software system. They define what the system should do, such as data processing, calculations, and specific business tasks.

Non-functional requirements describe how the system performs a function. This includes performance, scalability, security, usability, etc.

2. Explain why the testing phase is important in the SDLC, and provide two reasons for its significance.

Testing ensures that the developed software is free from defects and meets user expectations. It is a critical phase that helps to identify and fix bugs before deployment. Two main reasons:

- 1) It verifies the functionality of the software.
- 2) It improves the reliability and performance of the application.

3. Illustrate the concept of continuous integration in Agile Methodology and discuss its importance.

Continuous Integration (CI) is the practice of frequently integrating code changes into a shared repository. Each integration is verified by an automated build and testing process.

Importance:

- Helps in early detection of bugs.
- Reduces integration issues.
- Encourages collaboration and faster delivery.

4. Identify the key components of the Scrum framework and analyze how each contributes to effective project management.

Key components:

- Scrum Master: Ensures the team follows Agile practices and removes obstacles.
- Product Owner: Maintains the product backlog and prioritizes requirements.

- Development Team: Builds and delivers the product in incremental steps.
Together, these roles support transparent and flexible project development.

5. Evaluate the main steps involved in risk assessment and management, and assess their importance.

Steps:

- 1) Identify risks: Recognize potential issues that may affect the project.
- 2) Analyze risks: Evaluate likelihood and impact.
- 3) Plan risk response: Mitigate or eliminate risks.
- 4) Monitor and control: Track risks throughout the project.

Importance: Ensures the project is prepared to handle uncertainties, leading to successful delivery.

6. Explain the purpose of a Use Case Diagram in software development.

A Use Case Diagram represents the functional requirements of a system from the end-user perspective. It shows how users (actors) interact with the system and helps in identifying system functionalities early in development.

7. Compare and contrast a Sequence Diagram with an Activity Diagram.

Sequence Diagram shows the order of messages between objects over time. It focuses on time-based behavior.

Activity Diagram illustrates the workflow or business process. It focuses on control flow.

Key Difference: Sequence is time-oriented; Activity is logic-oriented.

8. Describe the Factory Pattern and explain how it differs from directly creating objects.

Factory Pattern provides an interface for creating objects in a superclass, but allows subclasses to alter the type of objects that will be created.

Difference: Instead of using 'new' to create objects directly, Factory Pattern uses a method to create them, offering better abstraction and flexibility.

Long Questions

1. Design a flowchart for a user registration process in a software application. Outline its key steps.

A user registration flowchart includes the following steps:

1. Start

2. User inputs name, email, password
3. System validates input (e.g., email format, password length)
4. Check if email already exists in database
5. If exists, show error; else, save user data
6. Display 'Registration Successful' message
7. End

This flowchart ensures that only valid and unique user data is stored.

2. Imagine you are managing a project to develop a simple mobile application. Describe how you would use the Agile Methodology to handle this project.

Agile Methodology divides the project into sprints. First, requirements are gathered and prioritized in the product backlog. During each sprint (1–2 weeks), the development team selects tasks to complete. Daily stand-up meetings help track progress.

At the end of each sprint, the working app version is shown to stakeholders for feedback. Changes are incorporated in future sprints.

This ensures adaptability, transparency, and continuous delivery of value.

3. You are working on a project that requires extensive documentation and has very specific regulatory requirements. Discuss why the Scrum methodology might not be suitable for this project and suggest an alternative methodology.

Scrum emphasizes flexibility, minimal documentation, and iterative development, which may not suit regulatory projects requiring strict compliance.

An alternative is the Waterfall Model, which provides a sequential and structured approach with extensive documentation at each phase.

Waterfall ensures clarity, accountability, and regulatory compliance, making it more appropriate for such projects.

4. Consider an online banking system. Create a Use Case Diagram to show the interactions between customers, bank staff, and the system.

Actors: Customer, Bank Staff

Use Cases: Login, View Balance, Transfer Funds, Deposit Money, Withdraw Money, Approve Loans

Customer interacts with login, view balance, transfer, etc.

Bank staff interacts with approve loans and monitor transactions.

This diagram helps in visualizing all interactions clearly for planning system features.

5. You are developing a food delivery application. Create a Sequence Diagram to show the process of placing an order, from the customer selecting items to the delivery of the order.

Sequence:

1. Customer selects food items
2. Places order through app
3. System processes order and notifies restaurant
4. Restaurant accepts and prepares food
5. System assigns delivery rider
6. Rider picks up and delivers food to customer
7. Order marked as completed

Each interaction is shown as a vertical timeline in the diagram.

6. Discuss the importance of software development tools in the software development process.

- a) Explain the role of language editors, translators, and debuggers in creating and maintaining software.**
- b) Provide examples of each tool and describe how they contribute to the efficiency and accuracy of software development.**

Software development tools improve productivity, quality, and maintainability of code.

a) Editors like Visual Studio Code are used to write and manage code efficiently. Translators like compilers (GCC, Java Compiler) convert source code into machine code.

Debuggers like GDB help find and fix bugs by allowing step-by-step code execution.

b) Editors: VS Code, Sublime Text

Translators: GCC, Python Interpreter

Debuggers: GDB, WinDbg

These tools automate and streamline the development process.