**University of KwaZulu-Natal**
**School of Engineering**
**Discipline of Electrical, Electronic, and Computer Engineering**
**(Howard College Campus)**
**Main Exam - November 2025**
**ENEL2SE H2: Software Engineering 1**

**Duration: Two hours** **Marks: 100**

Examiner Mr. M. ADAM
Moderator Dr. Sulaiman S. Patel

**Instruction:**

- Attempt all questions
- Start each question on a new page and number the answers.
- This paper must be kept in the exam hall.

## Question 1 [20 Marks]

Circle the letter of the correct answer.

1. Which architectural pattern is most suitable for a system that requires multiple, dynamic views of the same underlying data, such as a financial dashboard that displays both charts and tables? [2]
   a) Layered Architecture b) Model-View-Controller
   c) Client-Server d) Repository
2. Which software development model combines iterative development with systematic risk analysis in each cycle? [2]
   a) Waterfall Model b) Spiral Model
   c) V-Model d) Incremental Development

3. According to the "4+1" view model of software architecture, which view shows how the software is decomposed into interacting processes at runtime? [2]
   a) Logical View b) Development View
   c) Process View d) Physical View

4. The fundamental principle of Test-Driven Development (TDD) is to: [2]
   a) Write all the code first and then create tests to cover the functionality.
   b) Write tests after the code has been fully designed but before implementation.
   c) Write tests for a unit of functionality before writing the code that implements it.
   d) Focus only on writing tests for the user interface.

5. Which of the following is a key benefit of modularization in software design? [2]
   a) It enhances cohesion within individual modules, making them easier to maintain.
   b) It makes identifying reusable components more challenging.
   c) It focuses solely on external interfaces, ignoring internal structure.
   d) It increases coupling between different system components, reducing maintainability.

6. What is a primary goal of software testing? [2]
   a) To demonstrate that software meets requirements and to discover defects
   b) To eliminate all errors from the program
   c) To minimize code development time
   d) To increase system performance

7. Which testing technique involves examining individual program units, such as functions or object methods? [2]
   a) System testing b) Unit testing
   c) Component testing d) Integration testing

8. What is a significant advantage of using Agile development over the traditional Waterfall model, particularly in dynamic project environments? [2]
   a) Agile emphasizes thorough documentation before development begins.
   b) Agile supports iterative and incremental delivery, allowing continuous feedback and improvement.
   c) Agile follows a strict sequence of predefined phases without deviations.
   d) Agile development requires minimal customer involvement until the project is near completion.

9. Which is NOT a characteristic of a reasonable software requirement? [2]
   a) Correct b) Ambiguous c) Feasible d) Testable

10. What kind of error occurs when two subsystems operate at different speeds, and one accesses out-of-date information? [2]
    a) Interface misuse b) Interface misunderstanding
    c) Timing error d) Configuration error

**Question 2 [20 Marks]**

Match each term in List A with the most appropriate description from List B. Write the corresponding letter from List B next to the number in List A. [Each two marks].

| List A | | List B |
|---|---|---|
| 1. Agile Development | | A. A design pattern where an object notifies dependents of state changes. |
| 2. Cohesion | | B. An architectural pattern where all data is managed in a central repository. |
| 3. V-Model | | C. A development practice where tests are written before code. |
| 4. Integration Testing | | D. User testing where a release is made available to users for feedback. |
| 5. Observer Pattern | | E. An iterative development approach that emphasizes flexibility and customer collaboration. |
| 6. Beta Testing | | F. Testing the interactions between integrated components or systems. |
| 7. Verification | | G. A development model that extends the waterfall model with verification phases. |
| 8. Repository Pattern | | H. The measure of how closely related the functions within a module are. |
| 9. Test-Driven Development | | I. The high-level structure of a software system defines its components and relationships. |
| 10. Software Architecture | | J. The process of evaluating software to ensure it conforms to specifications. |

**Question 3[20 marks]**

Explain the key differences between the following pairs of concepts.

1. Waterfall Model vs. Agile Development (5 marks)
2. Black-Box Testing vs. White-Box Testing (5 marks)
3. Client-Server Architecture vs. Repository Architecture (5 marks)
4. Functional Requirements vs. Non-Functional Requirements (5 marks)

**Question 4 [40 marks]**

Read the scenario below carefully and answer the following questions.

The city council wants to develop a modern Public Transport Management System. The system will enable commuters to plan their journeys and pay for fares, while transport operators manage their vehicles and routes.

- Commuters can use a mobile app to search for routes, check real-time vehicle locations, purchase digital tickets, and view travel history. Before purchasing any ticket, the system must verify the commuter's account balance and check for any outstanding payments. This verification process is standard for all ticket purchases. If a vehicle is at full capacity, commuters cannot purchase tickets for that specific trip.

- Transport Operators can manage the system through a web dashboard. Their responsibilities include adding new routes, updating vehicle schedules, managing fare prices, monitoring vehicle locations in real-time, and generating performance reports.
- System Components include Mobile App Interface, Web Dashboard, Route Management component, Vehicle Tracking component, Payment Processing component, and Ticket Validation component. All components interact with a central Transport Database.

**Answer the following questions:**

1. Identify functional and non-functional requirements (8 marks)
    a) List 4 functional requirements
    b) List 4 non-functional requirements
2. Draw a Use-Case Diagram for the system. Your diagram must show the actors and use cases and correctly utilize <<include>> and <<extend>> relationships where appropriate. (5 marks)

3. Draw a Class Diagram for the system. Include at least five relevant classes (e.g., Commuter, Vehicle, Route, Ticket, Payment). For each class, show at least two attributes and one operation. Your diagram must illustrate: Association, Composition, and Inheritance relationships. (7 marks)

4. Draw a Sequence Diagram for the "Purchase Ticket" process. Include objects for: Commuter, MobileApp, PaymentProcessor, TicketSystem, and Database. Show the sequence of interactions from ticket selection to confirmation. (5 marks)

5. Draw a Component Diagram for the system architecture. Show the six main components described and illustrate dependencies between them. (5 marks)