Software Design and Development Mock Exam

Time: 2 hours 15 minutes

Instructions:

- Answer **all** 10 multiple choice questions in Section A.
- Answer **2 out of 3** questions in Section B.
- Each question in Section B is worth 30 marks.

Section A: Multiple Choice Questions (10 marks)

1. **Which of the following best describes module coupling?**

   a) The degree to which a module relies on other modules.

   b) The degree to which a module is cohesive.

   c) The number of lines of code in a module.

   d) The number of functions in a module.

2. **What is the primary goal of defensive coding?**

   a) To prevent errors from occurring in the first place.

   b) To make code more readable.

   c) To make code more efficient.

   d) To make code more secure.

3. **Which of the following is NOT a benefit of user testing?**

   a) Identifying usability issues.

   b) Gathering feedback on the design.

   c) Ensuring the code is error-free.

   d) Understanding user needs and preferences.

4. **What is the main purpose of version control?**

   a) To track changes to code over time.

   b) To prevent unauthorized access to code.

   c) To compile code into an executable file.

   d) To deploy code to a production environment.

5. **Which of the following is a key principle of test-driven development (TDD)?**

   a) Write code first, then write tests.

   b) Write tests first, then write code.

   c) Write tests and code simultaneously.

   d) Don't write any tests.

6. **What does the acronym DRY stand for in software development?**

   a) Don't Repeat Yourself

   b) Do Repeat Yourself

   c) Design, Review, Yield

   d) Develop, Refine, Yield

7. **Which SOLID principle aims to reduce the impact of changes by ensuring that classes are open for extension but closed for modification?**
   a) Single Responsibility Principle
   b) Open/Closed Principle
   c) Liskov Substitution Principle
   d) Interface Segregation Principle 1

8. **What is a common approach to achieving loose coupling between modules?**
   a) Using global variables.
   b) Defining clear interfaces.
   c) Tightly integrating modules.
   d) Sharing data directly between modules.

9. **In version control, what is a "branch" used for?**
   a) Deleting old code.
   b) Storing the main version of the code.
   c) Creating an independent line of development.
   d) Merging different versions of the code.

10. **Which of the following is NOT a type of software testing?**
   a) Unit testing
   b) Integration testing
   c) User acceptance testing
   d) Compilation testing

Section B: Open-Ended Questions (60 marks - answer 2 out of 3)

Question 1: Module Coupling and Cohesion (30 marks)

a) Explain the concepts of module coupling and cohesion. Provide examples of high and low coupling and cohesion. (10 marks)

b) Discuss the benefits of high cohesion and low coupling in software design. (10 marks)

c) You are designing a system for an online store. Identify three modules and explain how you would ensure low coupling and high cohesion between them. (10 marks)

Question 2: Defensive Coding and User Testing (30 marks)

a) Explain the importance of defensive coding in software development. Provide three examples of defensive coding techniques. (10 marks)

b) Describe the different stages involved in user testing. How can the results of user testing be used to improve software design? (10 marks)

c) A function calculates the total price of items in a shopping cart. Write this function in a language of your choice (JavaScript, Python, Pseudocode, or Flowchart) demonstrating defensive coding techniques to handle potential invalid inputs (e.g., negative quantity, invalid price). (10 marks)

Question 3: Version Control and Test-Driven Development (30 marks)

a) Explain the benefits of using version control systems like Git. Describe the key concepts of branching, merging, and committing in Git. (10 marks)

b) Outline the process of test-driven development (TDD). Discuss the advantages of using TDD in software development. (10 marks)

c) You are developing a function to validate a user's email address. Describe the steps you would take to develop this function using a TDD approach. You may use Javascript, Python, Pseudocode or a Flowchart to illustrate your tests and code. (10 marks)