Xi'an Jiaotong-Liverpool University
西交利物浦大学

Module Code	Examiner	Email of Examiner	Tel
CPT203			

Resit, 2024/25

Undergraduate – Year 3

Software Engineering 1

Exam Duration: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. **This is a closed-book examination, which is to be written without books or notes.**
2. **Total marks available are 100.**
3. **Answer all questions.**
4. **There is NO penalty for providing a wrong answer.**
5. **Students should write the answer on the booklet(s) provided.**
6. **Only English solutions are accepted.**
7. **All materials must be returned to the invigilator upon completion of the exam. Failure to do so will be deemed as academic misconduct and will be dealt with according to the University's policy.**

Question A (11 marks)

Suggest the most appropriate software process model that could be used as a basis for managing the development of the following systems, justify your suggestion:

A company plans to design a completely new driverless car to delivery heavy materials automatically. Its IT team is responsible for reliable software to control this car's operations, and make sure the safety of any person near the car.

A1 choose a software process model which is the best for this situation (3/11)

A2 provide the justification of your choice (8/11)

Question B (12 marks)

Compare agile development and waterfall model in the following three aspects:

B1 compare the differences of them in the scale of the system to be developed (3 /12)

B2 compare the differences of them in the detail level of documentation (3/12)

B3 compare the differences of them in interaction of the software development team (3/12)

B4 compare the differences of them in customer involvement (3/12)

Question C (12 marks)

Please compare the system requirements and user requirements, in the following aspects:

C1 compare their differences in technical specification (3/12)

C2 compare their differences in system architecture, design of software to be developed (3/12)

C3 compare their differences in the needs and perspectives of the end-users (3/12)

C4 compare their differences in functional and Non-Functional requirements (3/12)

Question D (15 marks)

You are tasked with creating a class diagram to model the system requirements for a study management system used by students and instructors. The system should manage courses, assignments, grades, and user accounts. The class diagram should capture the following requirements:

1. User: Represents a system user with attributes such as userID, name, email, and password. There are two types of users: Students and Instructors. Use inheritance to model these different types of users.

2. Course: Represents a course with attributes such as courseID, courseName, and description. A course can have multiple students enrolled and one instructor.

3. Assignment: Represents an assignment with attributes such as assignmentID, title, description, dueDate, and totalMarks. Each assignment is associated with one course.

4. Grade: Represents a grade with attributes such as gradeID, marksObtained, and feedback. A grade is associated with one assignment and one student.

5. Enrollment: Represents the enrollment of a student in a course with attributes such as enrollmentID and enrollmentDate. It is associated with one Student and one Course.

Please include the following relationships in your class diagram:

- A User can be either a Student or an Instructor.
- An Instructor can teach multiple courses, but each course is taught by one Instructor.
- A Student can enroll in multiple courses, and each course can have multiple students.
- A Course can have multiple assignments.
- An Assignment can have multiple grades, and each grade is associated with one Student and one Assignment.

Ensure that your class diagram includes:
D1 Classes with attributes. (5/15)
D2 Associations with appropriate multiplicities. (5/15)
D3 Generalization or specialization relationships where necessary. (5/15)

Question E (15 marks)
Imagine you are a software engineer working on a library management system. Your task is to create a state machine diagram to model the behavior of a "Book" within this system. The library management system is used by librarians and members to manage book loans, reservations, and returns.

Scenario:
When a book is first added to the library's collection, it starts in the "Available" state. The book can be checked out by the member, moving it to the "Checked Out" state. If the book is not returned by the due date, it transitions to the "Overdue" state. When a member returns the book, it moves to the "Returned" state. When an overdue book is returned, a fee will be charged to the borrower's account. After processing the return, the book goes back to the "Available" state.

Sometimes, books need maintenance or repair. A book can transition to the "Maintenance" state from the "Available" state if someone requests a repair action. Additionally, if a book is returned in bad condition, it transitions to the "Maintenance" state from the "Returned" state. If a book in "Maintenance" state is repaired, it transitions to the "Available" state. Finally, the book may transition from "Maintenance" to "Remove" if it is beyond repair. One month after it is transited into "Remove", its record will be permanently deleted from the system.

Your state machine diagram should include:
E1 All states mentioned above. (4/15)
E2 Transitions between states with appropriate triggers. (4/15)
E3 Entry and Exit when it is required. (4/15)
E4 An initial state and a final state. (3/15)

Question F (12 marks)

Suppose you are designing a simple online learning system that allows users to view courses, submit assignments, and participate in discussions. (12 marks)

F1 Explain what abstraction is and why it is important to use it in designing the system. (3/12)
F2 Using the assignment submission and course grading modules as examples, explain what high cohesion is and why it is beneficial in a modular system. (3/12)
F3 Identify two user interface design principles you would use to make the system easy to use. For each principle, explain how it would improve the user experience. (6/12)

Question G (11 marks)

You are supposed to do JUnit test for the following program (11 marks):

```
public class MathUtils {

    public int multiply(int a, int b) {
        return a * b;
    }

    public double squareRoot(double a) {
        if (a < 0) {
            throw new IllegalArgumentException("Cannot calculate square root of a
negative number");

        }
        return Math.sqrt(a);
    }
}
```

G1 Test case for multiplying two negative numbers and assert the result (3/11)
G2 Test case for calculating the square root of a negative number, ensuring the correct exception is thrown (4/11)
G3 Explain the purpose of the @BeforeAll and @AfterAll annotations in JUnit testing, and provide an example of how they can be used to manage a shared resource (4/11)

Question H (12 marks)

You are managing a software development project to deliver an e-commerce platform for a retail company. The project must be completed within a tight deadline and must ensure a high level of customer satisfaction with the product's performance and features.

H1 Identify and describe two major risks that could affect the success of this project, one from a project risk perspective and one from a product risk perspective. (4/12)

H2 For each risk, propose a risk management strategy: one strategy to avoid or mitigate the project risk and one strategy to reduce the impact of the product risk. (4/12)

H3 Provide the explanation of continuous improvement in software engineering, with an example of a technique used in this process. (4/12)

The end of the exam paper