

Module Code	Examiner	Email of Examiner	Tel
CPT203			

1st SEMESTER 2023/24 FINAL EXAMINATION

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、 This exam consists of two sections:

Section A consists of NINE questions for a total of 70 marks.

Section B consists of TWO systems modelling questions for a total of 30 marks.

Answer all questions. There is NO penalty for providing a wrong answer.
- 4、 Onsite candidate should write the answer on the booklet(s) provided.
- 5、 Only English solutions are accepted.
- 6、 All materials must be returned to the exam supervisor upon completion of the exam. Failure to do so will be deemed academic misconduct and will be dealt with accordingly.

Section A – Answer all questions below (70 marks):

Question A.1

Please provide a brief explanation of the four fundamental activities involved in software processes? (8 marks)

Question A.2

Compare the key differences between plan-driven processes and agile processes. (8 marks)

Question A.3

Please provide a diagram illustrating the flow of activities in the waterfall software process model. (6 marks)

Question A4

A management team is considering to invite an IT company to develop a new software to schedule the passenger lifts in tall buildings. It aims to increase the efficiency of operation during the peak time. Will you choose the agile method, as the leader of the IT company? Please provide three reasons to support your decision. (9 marks)

Question A5

List high-level activities in the requirement engineering process. (4 marks)

Question A6

Suppose you are developing a restaurant management system. Within this system, there is a class called *OrderProcessor* as shown below, responsible for handling customer orders. It has been identified that the code exhibits low cohesion. Your task is to rewrite the code in order to improve cohesion. (6 marks)

```
public class OrderProcessor {
    public void takeOrder(Order order) {
        // Logic for taking customer orders
    }

    public void prepareOrder(Order order) {
        // Logic for preparing orders in the kitchen
    }

    public void serveOrder(Order order) {
        // Logic for serving orders to customers
    }

    public void billCustomer(Order order) {
        // Logic for generating and presenting bills
    }
}
```

Question A7

Many web-based systems utilize MVC as the foundation for interaction management. Please provide a list of three components involved in this pattern and explain their respective functionalities. (6 marks)

Question A8

During the development of a basic user authentication system, you are required to create JUnit test cases to validate the functionality of the *UserAuthenticator* class. This class includes a method named *isValidPassword*, which verifies if a provided password satisfies specific criteria.

```
public class UserAuthenticator {
    public boolean isValidPassword(String password) {
        // Logic to check if the password meets certain criteria
        // For example, it must be at least 8 characters long
        // and contain at least one uppercase letter.
        return password.length() >= 8 && containsUppercase (password);
    }

    private boolean containsUppercase (String password) {
        for (char character : password.toCharArray()) {
            if (Character.isUpperCase(character)) {
                return true;
            }
        }
        return false;
    }
}
```

Write a JUnit test case, named *testIsValidPassword*, to evaluate the *isValidPassword* method functionality in the *UserAuthenticator* class. Ensure that the test covers both **valid** and **invalid** password scenarios. (11 marks)

Question A9

Please list and explain the three categories of strategies involved in the Risk Planning phase, and provide related examples to each of them. (12 marks)

Section B – Answer all requirements modeling questions below (30 marks):

Question B.1

A banking system provides users with several services:

- To check account balance. User can optionally check recent transactions;
- To pay bills online;
- To manage investments. When managing investments, user can buy and sell his investments.

All services require a secure log-in. The system must be secured, if the system crashes or the network connection fails during a transaction, all the financial data must be left in a consistent state. Draw a use-case diagrams for the requirements. (15 marks)

Question B.2

Produce a UML state machine diagram for a toy grabber machine with the following behaviors. (15 marks)

- The initial state of the toy grabber machine is Idle, where it waits for the player to insert coins.
- When a player is inserting coins, the machine enters the Collecting state. In this state, the machine shows the number of coins collected.
- In the Collecting state, the player has the option to press the cancel button at any time. If they choose to do so, all the inserted coins will be returned to them and the machine will return to the Idle state.
- If the player inserted 3 coins, the machine will play a song.
- After the song finishes, the machine enters the Accept Control state and begins accepting the player's control. Upon entering the Accept Control state, a timer starts counting down for 30 seconds.
- During the 30-second period in the Accept Control state, the system allows the player to control the machine and attempt to grab as many soft toys as they can.
- When the 30-second time limit is reached, the game comes to an end and the arm of the machine will reset to its original position.
- When the game ends, the machine will transition back to the Idle state.

The end of the paper