Sri Lanka Institute of Information Technology

B.Sc. Special Honours Degree/ Diploma

in

Information Technology

Final Examination

Year 2, Semester I (2023)

IT2020 - Software Engineering

Duration: 3 Hours

May/June 2023

Instructions to Candidates:

1. This paper contains **five** Essay Type Questions with a total of 100 marks.
2. Answer all questions in the booklet given.
3. This paper contains 8 pages with the Cover Page.
4. Electronic devices capable of storing and retrieving text, including calculators and mobile phones are not allowed.

Question 1 (25 marks)

Draw a sequence diagram for the following "Order Books" use case scenario.

Note: You may use suitable boundary, control, and entity classes.

Name: Order Books
Actor: Customer
Main Flow of Events:
1. Customer provides his/her username and password to LoginUI to login to the system.
2. The system validates the user.
3. If the valid login, show the "Login Success" message and directs it to the OrderUI.
4. First, the customer search for a book in the inventory using the "Search Book" option by entering the book details.
5. The details will be displayed to the customer if the book is available.
6. Customer adds the book to the cart.
7. At the same time, the inventory will be updated.
8. Customers can add any number of books to the cart following the same procedure (Steps 4 – 7).
9. Once the customer selects the "CheckOut" Option, the system will generate and display the total bill.
10. Then the customer can pay the bill online via a payment gateway.
11. Then he/she receives the order successful message.
Extension:
3a. If login unsuccessful, show "Invalid Login" message and ask to re-login.
5a. If the book is unavailable, display the "Unavailable" message.

Question 2 (20 marks)

Given below is a detailed description of an application developed for an online hotel reservation system called "QuickBooking". Model a **physical diagram** according to the given description.

This application can be used by both mobile and desktop users to reserve hotels. Web application users can access the system through a browser on a desktop or a laptop computer using "QuickBooking" web application while mobile users need to install the "QuickBooking" mobile application.

The main "QuickBooking" application runs in an Application Server installed in the IBM Server. Main QuickBook application contains three sub-components RoomSearch, Services, and Reservation. RoomSearch component allows to search or browse hotel rooms through iSearch interface to both desktop and mobile users. Also, the Reservation component allows to reserve hotel rooms via iReserve interface for both desktop and mobile users. The services component communicates with the Reservation and RoomSearch components via the iService interface implemented by the Services component.

The security and persistence components reside inside DellPowerEdgeServer. Security component is connected to RoomSearch, Service, and Reservation component via iEncryption interface implemented by itself. The persistence component communicates with the main "QuickBooking" application via iPersistence interface.

DB component is installed in a separate server. The persistence component uses the iConnector interface to communicate with the DB component. The application Server is connected with Desktop users and mobile users through HTTP protocol. Dell PowerEdge Server is connecting with IBM Server and Hardware Server through a wide area network.

Question 3 (25 marks)

a) Consider the partial code segment given below. Calculate the minimum number of test cases required for full statement coverage. (Show the statement coverage as a percentage for each test case) (15 marks)

```
boolean isPrimary, isSecondary;
int marks;
if (isPrimary)
{
      if(marks >=75)
      System.out.Println ("Excellent");
      else if (marks >=45)
      System.out.Println ("Good");
      else
      System.out.Println ("Try again");
}
else if(isSecondary)
{
     if(marks >=80)
     System.out.Println ("Excellent");
     else if (marks >=50)
     System.out.Println ("Average");
     else
     System.out.Println ("Repeat");
}
else
     System.out.Println ("Out of Scope");
```

b) Draw a control flow graph to the above code segment. (06 marks)

c) Considering above code segment, calculate the branch coverage for the scenarios given bellow. (04 marks)

i) isPrimary = False, isSecondary = True, marks = 75

ii) isPrimary = True, isSecondary = False, marks = 30

Question 4 (20 marks)

a)

In an ice cream shop, customers can request a basic ice cream and then add different toppings such as nuts, chocolate chips, sprinkles, and strawberries to it with whatever they choose. The added toppings alter the flavor of the plain ice cream. Also, the customers are free to mix and add many toppings as they wish.

i) Suggest the most suitable design pattern to implement this scenario and justify your answer.

(2 marks)

ii) Draw the relevant class diagram including appropriate methods which can be used for the given scenario.

(5 marks)

b)

A hotel has a hotel keeper. There are a lot of restaurants inside the hotel such as Veg restaurants, non-Veg restaurants, and Veg/Non-veg restaurants. A client wants to order from different menus of different restaurants. He/She does not know what the different menus have. So, they just have access to the hotel keeper who knows his hotel well. Whichever menu they want, they'll tell the hotel keeper, and he takes it out to the respective restaurants and hands it over to the client.

i) Suggest a suitable design pattern and justify your answer.

(2 marks)

ii) Illustrate the pattern using a class diagram.

(3 marks)

c) Consider the code given below and answer the questions.

(08 marks)

```
public interface Pen
{
  public void setColor(String color);
  public void draw(String content);
}
-------------------------------------------------------------------------------
public enum BrushSize {

  THIN, MEDIUM, THICK

 }
 -------------------------------------------------------------------------------
public class ThickPen implements Pen {

  final BrushSize brushSize = BrushSize.THICK;
  private String color = null;

  public void setColor(String color) {
    this.color = color;
  }

  public void draw(String content) {
    System.out.println("Drawing THICK content in color : " + color);
  }
}
 -------------------------------------------------------------------------------
public class ThinPen implements Pen {

  final BrushSize brushSize = BrushSize.THIN;
  private String color = null;

  public void setColor(String color) {
    this.color = color;
  }

  public void draw(String content) {
    System.out.println("Drawing THIN content in color : " + color);
  }
}
-------------------------------------------------------------------------------
public class MediumPen implements Pen {

  final BrushSize brushSize = BrushSize.MEDIUM;
  private String color = null;

  public void setColor(String color) {
    this.color = color;
  }
  public void draw(String content) {
    System.out.println("Drawing MEDIUM content in color : " + color);
  }
```

```
}
-------------------------------------------------------------------------
import java.util.HashMap;

public class PenF
{
  private static final HashMap<String, Pen> pensMap = new HashMap<>();

  public static Pen getThickPen(String color)
  {
    String key = color + "-THICK";

    Pen pen = pensMap.get(key);

    if(pen != null) {
      return pen;
    } else {
      pen = new ThickPen();
      pen.setColor(color);
      pensMap.put(key, pen);
    }

    return pen;
  }

  public static Pen getThinPen(String color)
  {
    String key = color + "-THIN";

    Pen pen = pensMap.get(key);

    if(pen != null) {
      return pen;
    } else {
      pen = new ThinPen();
      pen.setColor(color);
      pensMap.put(key, pen);
    }

    return pen;
  }

  public static Pen getMediumPen(String color)
  {
    String key = color + "-MEDIUM";

    Pen pen = pensMap.get(key);

    if(pen != null) {
      return pen;
    } else {
      pen = new MediumPen();
```

```
      pen.setColor(color);
      pensMap.put(key, pen);
    }

    return pen;
  }
}
-------------------------------------------------------------------------
public class PaintBrushClient
{
  public static void main(String[] args)
  {
    Pen yellowThinPen1 = PenF.getThickPen("YELLOW");
    yellowThinPen1.draw("Hello World !!");

    Pen yellowThinPen2 = PenF.getThickPen("YELLOW");
    yellowThinPen2.draw("Hello World !!");

    Pen blueThinPen = PenF.getThickPen("BLUE");
    blueThinPen.draw("Hello World !!");

    System.out.println(yellowThinPen1.hashCode());
    System.out.println(yellowThinPen2.hashCode());

    System.out.println(blueThinPen.hashCode());
  }
}
```

i) Identify the design pattern used in the code. (01 mark)

ii) What is the main purpose of using the design pattern you have mentioned above for this solution? (01 mark)

iii) Draw the class structure of the design pattern that you identified in part (i) with appropriate methods for the above scenario. (06 marks)

Question 5 (10 marks)

a) Explain three types of triggers in a state machine diagram with suitable examples.

(03 marks)

b) Explain the difference between Code line and Base line in Version Management.

(02 marks)

c) What is an incident in incident management? (02 marks)

d) Explain Incident Response in Incident Management (03 marks)

---------- END OF THE QUESTION PAPER ----------