00108

Student ID:

Machine No:

Sri Lanka Institute of Information Technology

B.Sc. Honours Degree in Information Technology

Specialized in Information Technology

Final Examination (Computer Based)
Year 2, Semester 1 (2023)
Paper Version B

IT2030 – Object Oriented Programming

Duration: 3 Hours

November 2023

Instructions to Candidates:

- This paper contains four questions. Answer All Questions.
- Marks for each question are given on the paper. Total Marks: 100.
- Create a folder on the Desktop with your student registration number and store all your program files inside that.
- Create a separate Project for each question. The name of the project is provided in the question.
- This paper contains 06 pages including the Cover Page.

Instructions to Candidates when submitting:

- Save all your work.
- Delete all unnecessary files from each project folder (There should be 4 folders for 4 projects named as Question01, Question02, Question03 and Question04 inside your ID folder, and each folder should contain the source code (.java files)) only.
- Zip the Student ID folder (Zipped folder also should be named with Student ID number).
- Upload the zipped folder to the correct link.

Question 1 (20 marks)

This question is based on the **OOP concepts**.

Implement the necessary classes, along with their required attributes and methods, based on the following description.

You have been assigned to develop a customer order processing system for an online store. The system should consist of two classes, **Order** and **Product**. Each class should have specific attributes, methods, and functionalities as follows.

The **Order** class should have an order ID (int), customer name (String), order date (String), and a queue to hold the selected products. The class should include a default constructor, an overloaded constructor to initialize these attributes. Additionally, you are required to implement the two methods below:

- AddProduct(Product p): This method should allow the addition of selected products to the order queue.
- ProcessOrder():This method should display the order details and the list of products ordered.

The **Product** class should have a product ID (int), name (String), price (double), and stock quantity (int). It should include a default constructor and, an overloaded constructor to initialize these attributes, and two methods, Read() and Display(). The Read() method should accept inputs for the above-mentioned attributes through user inputs, and the Display() method should output their values.

Implement a class called "**OrderProcessingApp**" with the main method. In the main method, create 4 instances of the Product class and 2 instances from Order class. Input order details using the overloaded constructor for the Order class and product details using the Read() method of the Product class. Add products to the order using the AddProduct() method. Display the order details and the list of products in the order using the ProcessOrder() method.

Save the project as Question01

Question 2 (30 marks)

This question is based on the **Collection Framework and Generics**.

a) Write a Java program that allows the user to manage a list of employee names using ArrayList. The program should perform the following tasks.

i) Create an empty ArrayList of Strings to store employee names. Then Prompt the user to enter the names of employees one by one. Keep accepting names as keyboard inputs until the user enters "done". Add each entered name to the created ArrayList. (04 marks)

ii) Once the user finished entering the names, ask the user to enter the name of the employee they want to remove. If the employee's name is found in the ArrayList, remove it. Display a message indicating whether the name was successfully removed or not in the case if the name was not found. (06 marks)

iii) Find and display the total number of employees in the ArrayList. (02 marks)

iv) Print all the remaining employee names in the ArrayList. (03 marks)

Save the project as **Question02A**

b) Create a class called "**GenericCalculator**" with two generic methods as follows.

i) Define another generic method called **CalculateMaximum()** that takes an array of a numeric data type (Integer, Double, Float) as input and returns the maximum value from array. (05 marks)

ii) Define a generic method called **calculateProduct()** that takes an array of a numeric data type (Integer, Double, Float) as input and returns the product of its elements. This method should multiply all the elements in the array and return the result. (05 marks)

iii) Write a class called "**MainApp**" which contains the main method. Create two objects from the **GenericCalculator** class where the first object accepts an Integer type array for the above two methods and the second object accepts a Double type of array for the above two methods. Display the results using proper printing statements. (05 marks)

Save the project as **Question02B**

Question 3 (20 marks)

This question is based on the **Threads implementation**.

You have two types of vehicles, "**Car**" and "**Bike**," that need to be tested on a racetrack concurrently using Java threads.

a) Create a class called "**Car**" which can simulate racing by displaying "Car is racing" five times, each time followed by a 500-millisecond delay to simulate a racing lap. (06 marks)

b) Create another class called "**Bike**" which can simulate racing by displaying "Bike is racing" five times, each time followed by a 500-millisecond delay to simulate a racing lap. (06 marks)

c) Implement another class called " **RaceRunner**" with a main method. In the main method, create one thread object each from **Car** and **Bike** classes. Start both threads to simulate the concurrent racing of a car and a bike. (05 marks)

d) In the execution ensure that the car completes its race before the bike starts. (03 marks)

Hint: Refer to the sample output given below

Save the project as **Question03**

Sample Output:

```
Car is racing
Car is racing
Car is racing
Car is racing
Car is racing
Bike is racing
Bike is racing
Bike is racing
Bike is racing
Bike is racing
```

Question 4 (30 marks)

This question is based on the **Design Patterns** implementation.

a) You are going to implement the **Strategy Design Pattern** based on the scenario of **Robots** usage for **Textile industry** for different patterns designing purpose in Japan and China. These robots can design either lines (Dash lines, dotted lines, and Straight lines) or shapes (Square, Circle, and Triangle).

i). Implement two interfaces **IDrawableLines** and **IDrawableShapes** and each interface you should declare methods **drawLines(String robotType)** or **drawShapes(String robotType)**

(02 marks)

ii). Then create 3 classes to design lines as **DashLines**, **StraightLines**, and **DottedLines** and those classes should implement the **IDrawableLines** interface and override all methods within the classes.

(06 marks)

iii). Similarly create another 3 classes **SquareShapes**, **TriangleShapes**, and **CircleShapes** and those classes should implement the **IDrawableShapes** interface and override the method as well.

(03 marks)

iv). Create an **Abstract** class **DesignSelector** and aggregate two interfaces **(IDrawableLines,** and **IDrawableShapes),** you should set those two behaviors with using "set" methods and let design lines or shapes to make the decision in sub classes

(07 marks)

b) Now design two robots from two different countries they can let to show robot type either **Japanese** or **Chinese** and accordingly it should display the outputs as given in the **TextileIndustryDemo**.

i). Now implement the **JapaneseRobot** and **ChineseRobot** classes to draw lines or shapes according to that format and extend the **DesignSelector** class in each to implement those behaviors.

(06 marks)

ii). Now modify all line classes **(DashLines, StraightLine, DottedLine)** and all the shape classes **(SquareShape, CircleShape, and TraingleShape)** according to the **Singleton Pattern**

(06 marks)

Hint: Refer to the main program and package hierarchy given below

Save the project as **Question04**

Main Program:

```
1  package oop.exam.q2;
2
3  public class TextileIndustryDemo {
4
5      public static void main(String[] args) {
6
7          DesignSelector japaneseRobot = new JapaneseRobot();
8          japaneseRobot.[redacted](DashLines.getInstance());
9          japaneseRobot.designLines();
10         japaneseRobot.[redacted](DottedLines.getInstance());
11         japaneseRobot.designLines();
12         japaneseRobot.[redacted](StraightLines.getInstance());
13         japaneseRobot.designLines();
14
15         DesignSelector chineseRobot = new ChinesRobot();
16         chineseRobot.setiDrawableShapes(CircleShapes.getInstance());
17         chineseRobot.designShapes();
18         chineseRobot.setiDrawableShapes(SquareShapes.getInstance());
19         chineseRobot.designShapes();
20         chineseRobot.setiDrawableShapes(TriangleShapes.getInstance());
21         chineseRobot.designShapes();
22     }
23 }
```

```
Console
<terminated> TextileIndustryDemo [Java Application] C:\Progra
Draw Dash Lines on clothes by Japanes Robot
Draw Dotted Lines on clothes by Japanes Robot
Draw Straight Lines on clothes by Japanes Robot
Draw Circles on clothes by Chinese Robot
Draw Squares on clothes by Chinese Robot
Draw Triangles on clothes by Chinese Robot
```

Package Hierarchy:

- oop.exam.q2
  - ChinesRobot.java
  - CircleShapes.java
  - DashLines.java
  - DesignSelector.java
  - DottedLines.java
  - IDrawableLines.java
  - IDrawableShapes.java
  - JapaneseRobot.java
  - SquareShapes.java
  - StraightLines.java
  - TextileIndustryDemo.java
  - TriangleShapes.java