00463

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 (2024)
Paper Version B

IT2030 – Object Oriented Programming

Duration: 3 Hours

June 2024

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 (25 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 are required to develop a student management system for a university which consists of two classes: **Student** and **StudentManager**.

The **Student** class should have a studentID (int), name (String), age (int), major (String), and GPA (double) as attributes. This class should include a default constructor and an overloaded constructor to initialize these attributes. Additionally, the following methods need to be implemented:

- **updateGPA(double newGPA)**: This method should update the available GPA of the student to the new GPA.
- **displayStudentDetails()**: This method should display all the details of the student.

The **StudentManager** class should have an ArrayList called studentList to store a collection of student objects. This class should include a default constructor and the following methods:

- addStudent(Student student): This method should add a student to the student list.
- searchStudent(int studentID): This method should search for a student by their ID and display their details.
- updateStudentGPA(int studentID, double newGPA): This method should update the GPA of a student by their ID.
- displayAllStudents(): This method should display the details of all the students in the student list.

Implement a class called "**StudentManagementApp**" with the main method. In the main method, create an object of the **StudentManager** class. Populate the student list with at least 5 students using the addStudent() method. Perform the following operations:

- Search for a student by ID using the searchStudent() method.
- Update the GPA of a student using the updateStudentGPA() method.
- Display the details of all the students in the student list using the displayAllStudents() method.

Save the project as **Question01**.

Question 2 (30 marks)

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

a) Develop a Java program for managing a Product Inventory System using **HashMap**. The program should perform the following tasks:

i). Create an empty HashMap to store product records. Then prompt the user to enter the details of products one by one. Accept inputs for product ID (int), and quantity (int) as keyboard inputs. Keep accepting inputs until the user enters "-99". Add each entered student record to the HashMap. (04 marks)

ii). Once the user finishes entering the student records (once the user enters "-99"), ask the user to enter the id of the product that needs to be removed. If the product id is found in the HashMap, remove the corresponding product record. Display a message indicating whether the record was successfully removed or not in the case the product ID was not found. (06 marks)

*[Hint: Use **containsKey(id)** to find a key in the HashMap]*

iii). Find and display the total number of products in the HashMap. (02 marks)

iv). Print all the remaining product records in the HashMap. (03 marks)

Save the project as **Question02A**.

b) Implement a Java program to demonstrate the use of **generics** in a class called "**GenericInventory**". The program should accomplish the following tasks:

i). Define a generic method called "**calculateAverage()**" that takes an array of a numeric data type (Integer, Double, Float) as input and calculates the average value of the elements in the array. (05 marks)

*[Hint: convert the array elements to double values using **doubleValue()** method]*

ii). Define another generic method called "**calculateMinimum()**" that takes an array of a numeric data type (Integer, Double, Float) as input and displays the minimum value from the array. (05 marks)

iii). Write another class called "**InventoryManager**" which contains the main method. Create two objects from the "**GenericInventory**" 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**.

Implement two classes called "**Telescope**" and "**Satellite**" to simulate their observational routines using Java threads.

a) Create a class called "**Telescope**" which simulates a telescope observing the night sky by displaying "**Telescope** is observing" five times, with each message followed by a 500-millisecond delay to simulate time spent on each observation. (04 marks)

b) Create another class called "**Satellite**" which behaves similarly to the Telescope class, displaying "Satellite is observing" five times, each message followed by a 500-millisecond delay to simulate time spent on each observation. (04 marks)

c) Implement a class called "**ObservationControler**" with a main method. In this method, create one thread object each for the Telescope and Satellite classes. Start both threads to simulate the concurrent observations of both celestial bodies. (04 marks)

d) Ensure that the Telescope completes all its observations before the Satellite begins its observational sequence. Manage this order of execution using appropriate synchronization techniques within the program. (03 marks)

e) After the Telescope and Satellite have completed their initial observations, modify the **ObservationControler** class to allow both observations again simultaneously. (05 marks)

[Hint: Refer to the sample output given below]

Save the project as **Question03**.

Sample Output:

```
Telescope is observing
Telescope is observing
Telescope is observing
Telescope is observing
Telescope is observing

Satellite is observing
Satellite is observing
Satellite is observing
Satellite is observing
Satellite is observing


Both observations begin again simultaneously:
Telescope is observing
Satellite is observing
Satellite is observing
Telescope is observing
Satellite is observing
Telescope is observing
Satellite is observing
Telescope is observing
Satellite is observing
Telescope is observing
```

Question 4 (25 marks)

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

You are required to design and implement a multimedia control system using the **Command** Pattern and the **Singleton** Pattern. Implement two classes called **Television** and **SoundSystem** as receivers with specific functionalities. Define a **Command** interface along with several concrete command classes to control the device operations. Implement a class called **RemoteControl**, accommodating the singleton pattern to manage these commands. The client setup is given in the **MultimediaControlTest** class, simulating button presses to execute commands on the devices.

*[Hint : Create a java project called "**Question04**". Download the zip folder given in the netexam page and extract it. Then copy all the java files to the created project in the Eclipse environment (Question04). The Client programme code and a sample output is given for the reference.]*

Follow the below steps to complete the system.

a) Implement following methods within both the receiver classes called "**Television**" and "**SoundSystem**".

   i). **Television**: Implement 3 methods called "turnOn()", "turnoff()" and "setChannel(int channel)" with appropriate display statements. (03 marks)

   ii). **SoundSystem**: Implement 3 methods called "turnOn()", "turnoff()" and "increaseVolume()" with appropriate display statements. (03 marks)

b) Implement the command design pattern as follows.

   i). Define a method called "execute()" within the **Command** interface. (02 marks)

   ii). Implement the above interface in **TvOnCommand**, **TvOffCommand**, **SetChannelCommand**, **SoundOnCommand**, **SoundOffCommand**, **IncreaseVolumeCommand** classes. (12 marks)

c) Implement the Singleton design pattern within the **RemoteControl** class as follows.

   i). Within the **RemoteControl** class define an array of Command objects. This array will store the commands assigned to each button on the remote. (01 marks)

   ii). Implement a method called "setCommand(int slot, Command command)" that assigns a command to a specific button slot in the remote control. (02 marks)

iii). Implement a method called "pressButton(int slot)" which checks if a command is assigned to the specified button slot and, if so, executes the command by calling it execute() method. This simulates pressing a button on the remote control.

(02 marks)

Save the project as **Question04**

Sample Output:

```
TV turned on
TV channel set to 101
Sound system turned on
Sound system volume increased
```