Sri Lanka Institute of Information Technology

B. Sc. Honours Degree

in

Information Technology

Final Examination (Computer Base)

Year 2, Semester I (2022)

CSIT2030 – Object Oriented Programming

Duration: 3 Hours

Instructions to Candidates:

- This paper contains **Four** questions. **Answer All** Questions.
- Write your student ID on top of the paper.
- Marks for each question are given in the paper.
- Write your student ID on top of the exam paper.
- Total Marks: 100.
- Create a separate Project for each question. The name of the project is provided. Save each Java program using the class name given.
- Store all your program files in the Desktop Folder provided
- This paper contains 05 pages with the Cover Page.

Instructions to Candidates when submitting:

- Save all your work.
- Create a folder from your student ID.
- Inside that, create 4 separate folders from the project name provided.
- Copy each project answer source codes(Only the .java files) in to respective folders.(There should be 4 folders name as **Question01A, Question02A, Question03A** and **Question04A** inside your ID folder, and in each folder should contain the answer.(.JAVA files ONLY).
- Zip the Student ID folder (Zip folder also should be the Student ID number).
- Upload into the correct link.

Question 1 (30 marks)

This question is based on the **Object-Oriented Programming (OOP) concepts**.

You are going to implement a java code for a game called "PetGame" mobile game. PetGame rules are simple, adopt pets and take good care of them, and gain points! Beware of the pets from other carnivore pets.

PointAllocator is an interface where it has a getter and a setter for the points. Pet is a class that use the PointAllocator to set and get points for each pet. The real implementation of the getter and setter is implemented in the pet class. Apart from these methods, there are some other methods as feed, clean, cuddle, GetTotalPoints which the real implementation of these can be done when only knowing the exact pet type. There are 3 types of pets as Cat, Squirrel and Parrot. In each class, feed, clean and cuddle will display a suitable message and pointes added respectively as 5, 10 and 15. Squirrel and Parrot have an additional attribute called command, where used in method called isCaught. In the Squirrel class, if the command given for the squirrel to "Run" for "5" times, isCaught should generates a custom exception called CaughtException , and catch and handle it and return true. Else, display a suitable message and return false. In the Parrot class, if the command given for the parrot to "Walk", isCaught should generates the custom exception called CaughtException , and catch and handle it and return true. Else, display a suitable message and return false.Accordingly, in both Squirrel and Parrot class methods, GetTotalPoints should be implemented where if the pet is caught, points should be zero, else, total points will be displayed. Cat class GetTotalPoints will display the total points.

Refer to the output given below and implement the necessary classes and methods in java.

```
3  public class Demo {
4      public static void main(String[] args) {
5          Pet squirrel1 =new Squirrel("Run3");
6          squirrel1.cuddle();
7          squirrel1.feed();
8          squirrel1.GetTotalPoints();
9
10         Pet squirrel2 =new Squirrel("Run5");
11         squirrel2.cuddle();
12         squirrel2.feed();
13         squirrel2.GetTotalPoints();
14
15
16         Pet garfield =new Cat();
17         garfield.clean();
18         garfield.cuddle();
19         garfield.feed();
20         garfield.GetTotalPoints();
21
22
23         Pet parrot1 =new Parrot("walk");
24         parrot1.feed();
25         parrot1.GetTotalPoints();
26     }
27 }
```

```
<terminated> Demo (5) [Java Application] C:\Progra
Cuddling the Squirrel
Feeding the Squirrel
Did not caught
Total point is 20.0
Cuddling the Squirrel
Feeding the Squirrel
Q1.CaughtException: Caught
Cleaning the Cat
Cuddling the Cat
Feeding the Cat
Total point is 30.0
Feeding the Parrot
Q1.CaughtException: Caught
```

Save the project as **Question01A**

Question 2 (15 marks)

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

1) Implement a generic class that can store Numerical type data called TCalculation.
   a. Include a property called numList which is an ArrayList type variable. It should be able to store only the generic Numerical type values.

      (02 marks)

   b. Include a method called append which takes a generic type parameter and appends the values to the numList and returns nothing.

      (04 marks)

   c. Include another method called average which calculate the average of all the values stored in the numList and return the average.

      Hint: The number class has a method called doubleValue()

      (04 marks)

2) Implement a class called DemoApp with the main method to demonstrate the behaviors implemented above.
   a. Create TCalculation type object which stroes only int type data.
   b. Create another TCalculation type object which stores double type data.
   c. Store some sample values and calculate and print the average of the values.

      (05 marks)

Save the project as **Question02A**

Question 3 (25 marks)

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

1) There is a **producer – consumer scenario,** where you should provide the solution using **Threads**. You should create two Threads as **ProducerThread** and **ConsumerThread** those execute sequentially and put elements to a queue (you can use an ArrayList). The **ProducerThread** should add elements to the queue and the **ConsumerThread** should consumes elements from the queue and each time queue maintain the **empty elements state**. Each time when you execute a thread there is **1 second time interval** and both threads behave in **synchronized** manner.
[Hint: - To implement this Program you should use wait(), notify() and sleep() methods.]

Refer the below sample java code of the Main program and check the Console Outputs. Write a java code to display the same output.

```
3  import java.util.ArrayList;
4
5  public class ThreadTest {
6
7      public static void main(String[] args) {
8          ArrayList<Integer> queue = new ArrayList<>();
9          Thread producer = new Thread(new ProducerThread(queue));
10         Thread consumer = new Thread(new ConsumerThread(queue));
11         producer.start();
12         consumer.start();
13         System.out.println("Starting");
14     }
15 }
16
17
18
19
20
21
22
23
24
25
26
```

```
Producer started
Starting
Producer adding value = 10 to Queue
Consumer started
Consumer thread consumes 10
Elements in Queue = []
Producer started
Producer adding value = 20 to Queue
Consumer started
Consumer thread consumes 20
Elements in Queue = []
Producer started
Producer adding value = 30 to Queue
Consumer started
Consumer thread consumes 30
Elements in Queue = []
Producer started
Producer adding value = 40 to Queue
Consumer started
Consumer thread consumes 40
Elements in Queue = []
Producer started
Producer adding value = 50 to Queue
Consumer started
```

a) Create the **ProducerThread** and access the Queue and add elements. Producer should add elements by incrementing 10 by 10. Thread should work as an infinite loop.

(13 marks)

b) Create the **ConsumerThread** and it should remove elements from the queue, and it also should work as an infinite loop.

(12 marks)

Save the project as **Question03A**

Question 4 (30 marks)

This question is based on the **Design Patterns**. You have to implement the **Bridge** design pattern.

"AutoCar" is a well-known auto mobile company which produces and assembles products for cars. The company manufacturing different types of items like Air Bags or Automatic Braking System(ABS). They get the car, produce the system for it and assemble it into the car.
Recently, they got new orders from "Toyota" to produce front Air bag and side Air bag system for their new car model.
After a while, another car company, "Nissan" asked them to produce front Air bag and side Air bag system for their model. Thinking if another car company demands another front Air bag and side Air bag, AutoCar has decide to implement a software system which the car and the product is vary independently so that it is easy to extend and reuse.
To maintain this, they are creating a new software system.In the **CarAirBag**, in which they kept some car airbag specific methods like **airBagMotionDetection** and **airBagLightIndecator** which common to all airbag products. Both **FrontAirBag** and **SideAirBag** has same structures across the CarAirBag including same functionalities. The Car interface provides two abstract functionalities as **assembleLight** and **assembleMotionSensor**. **ToyotaCar** and **NissanCar** now use the Car to assemble the airbag light and the motion Sensor.

Refer to the below sample code of the Main program and Console Outputs. Your java program also should display the same output.

```
2
3  public class Demo {
4
5      public static void main(String[] args) {
6          CarAirBag fAirbag = new FrontAirBag();
7          CarAirBag sAirbag = new SideAirBag();
8
9          new NissanCar(fAirbag).assembleLight();
10         new NissanCar(fAirbag).assembleMotionSensor();
11         new NissanCar(sAirbag).assembleMotionSensor();
12
13         new ToyotaCar(fAirbag).assembleLight();
14         new ToyotaCar(fAirbag).assembleMotionSensor();
15         new ToyotaCar(sAirbag).assembleLight();
16         new ToyotaCar(sAirbag).assembleMotionSensor();
17     }
18
19 }
```

```
Assembling for Nissan
Light Indicator on for Front Air Bag
Assembling for Nissan
Motion detection on for Front Air Bag
Assembling for Nissan
Motion detection on for Side Air Bag
Assembling for Toyota
Light Indicator on for Front Air Bag
Assembling for Toyota
Motion detection on for Front Air Bag
Assembling for Toyota
Light Indicator on for Side Air Bag
Assembling for Toyota
Motion detection on for Side Air Bag
```

Save the project as **Question04A**

End of The Examination Paper