00125

Student ID:	
Machine No:	

Sri Lanka Institute of Information Technology

B.Sc. Honours Degree in Information Technology

Specialized in Information Technology

Final Examination (Computer Base)
Year 2, Semester 1 (2022)
Paper Version A

IT2030 – Object Oriented Programming

Duration: 3 Hours

November 2022

Instructions to Candidates:

- This paper contains four questions. Answer All Questions.
- Marks for each question are given in the 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 08 pages including 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 Question01, Question02, Question03 and Question04 inside your ID folder, and in each folder should contain the answer (.JAVA files ONLY)).
- Zip the Student ID folder (Zipped folder also should be named with Student ID number).
- Upload the zipped folder into the correct link.

Question 1 (25 marks)

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

a) Implement an abstract class called **Employee** to do the following.

i) Store the following properties
id (int), name (String)
(01 mark)

ii) Implement a constructor to get two properties as parameters and initialize them.
(02 marks)

iii) Implement a method called **display()** to display these two properties.
(01 mark)

iv) Define an abstract method called **calculateNetSalary().**
(01 mark)

b) Implement a sub class of Employee class called **PermanentEmp** to do the following.

i) Store the following properties
basicSal (double), commission (double)
(01 mark)

ii) Implement a constructor to get three properties (id, name and basicSal) as parameters and initialize them. The **commission** should be assigned as 15% from the basic salary.
(02 marks)

iii) Override the **calculateNetSalary()** method to calculate net salary of the employee.
[Hint: Net Salary = Basic Salary + Commission]
(02 marks)

iv) Implement a method called **display()** to display all the properties of the class.
[Please refer to the below screenshot of the expected output]
(02 marks)

c) Implement another sub class of Employee class called **TemporaryEmp** to do the following.

i) Store the following properties
OtHrs (int), OtRate (double), minWage (double)
(01 mark)

ii) Implement a constructor to get four (id, name, OtHrs and OtRate) properties as parameters and initialize them. The **minWage** should be assigned as 15000/- for each temporary employee.
(02 marks)

iii) Override the **calculateNetSalary()** method to calculate net salary of the employee.
[Hint: Net Salary = Minium Wage + OT Amount]

(02 marks)

iv) Implement a method called **display()** to display all the properties of the class.
[Please refer to the below screenshot of the expected output]

(02 marks)

d) Implement a class called **DemoApp** with main() method to do the followings.

i) Create an ArrayList of **PermanentEmp** class and add two **PermanentEmp** objects to the arraylist.

(02 marks)

ii) Call the **calculateNetSalary()** and display() method for both objects using foreach loop.

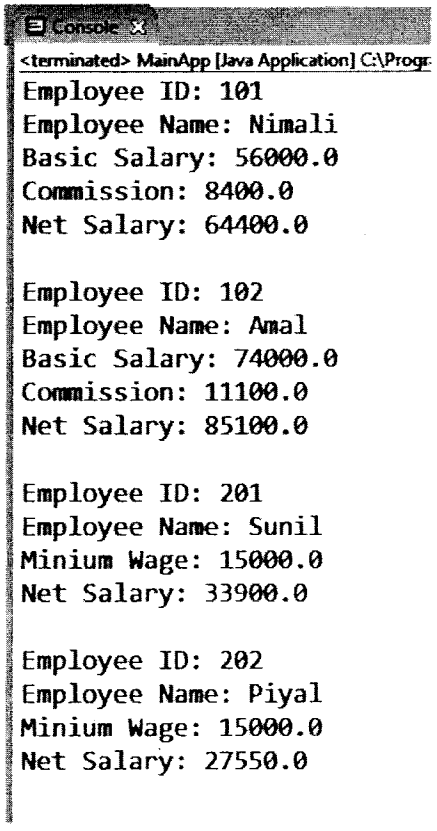
(02 marks)

iii) Do the part (i) and part (ii) above for the **TemporaryEmp** class
[Please refer to the below screenshot of the expected output]

(02 marks)

Save the project as **Question01**

Sample Output:



Question 2 (15 marks)

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

a) Implement a generic class called **Book**.

i) Include two properties called **bookTitle**, and **bookISBN**. These properties should be able to hold any data type. For example, **bookISBN** can be a String or even an Integer value.

(04 marks)

ii) Include an overloaded constructor that takes in two parameter to initialize the two properties.

(03 marks)

iii) Include another method called **getISBN()** that will return the **bookISBN** of the Book object.

(02 marks)

b) Implement a class called **Library** with the **main** method.

i) Create a **HashMap** object called **bookList** that uses an Integer value as the Key and a **Book** object as the value.

(02 marks)

ii) Create a **Book** object with the following values as the **ISBN** and **Title** and add to the **bookList**

bookISBN: "ISBN12345"

bookTitle: "Harry Potter"

(2 marks)

iii) Create another **Book** object with the following values as the **ISBN** and **Title** and add to the bookList.

bookISBN: 123456

bookTitle: "Rings of Power"

(2 marks)

Save the project as **Question02**

Refer to the partial code of the main method given below.

```
class Main {
  public static void main(String[] args) {
    //start your code here

    //End your code here
    for (int id: bookList.keySet()) {
      Book value = bookList.get(id);
      System.out.println("Book ID is: "+ id + "and the ISBN is: " + value.getISBN());
    }
  }
}
```

Question 3 (30 marks)

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

A Wrapping paper art is printed using computer program and which is drawn using two concurrent Threads. You are allowed to enter pattern styles through keyboard inputs and you should select number of occurrences (count) the style to be printed. Each thread should print patterns one after the other and you should print the triangle shape using given style. Set names for the Threads as **Pattern01**, and **Pattern02**.

Refer the sample output.

*[Assumption: - **Thread synchronization** is essential and both threads should print the output as synchronized manner. Correct implementation of **wait(), notify()** methods is compulsory to obtain full marks]*

a) Implement the **Pattern01** class as below.
   i) Overload the **Pattern01** constructor with a lock (for synchronization), and use Triangle **pattern** (String), **count** (int) as parameters.
   (01 mark)
   ii) Override the **run()** method and implement the pattern print logic.
   (10 marks)
   iii) In each iteration the Thread should sleep for **1 second** of time interval and it should print the thread name and given values as per the given output.
   (01 mark)

b) Implement the **Pattern02** class as below
   i) Overload the **Pattern02** thread constructor with a lock (for synchronization), and use Art work **pattern** (String), and **count** (int) as parameters.
   (01 mark)
   ii) Override the **run()** method and implement the pattern print logic
   (10 marks)
   iii) In each iteration the Thread should sleep for **1 second** of time interval and it should print the thread name and given values as per the given output.
   (01 mark)

c) Implementation the **Main** method as below

i) Use Keyboard inputs to get the Pattern styles

(03 marks)

ii) Implement two Threads and set names **Pattern_01**, and **Pattern_02 Thread** and pass necessary parameters

(03 marks)

Save the project as **Question03**

Sample Output:

```
Enter Pattern 1 = +
Enter Pattern 2 = -
Enter count = 8
=======Threads start printing patterns.=======
Pattern 02 Thread =          -
Pattern 01 Thread =          +
Pattern 02 Thread =         - -
Pattern 01 Thread =         + +
Pattern 02 Thread =        - - -
Pattern 01 Thread =        + + +
Pattern 02 Thread =       - - - -
Pattern 01 Thread =       + + + +
Pattern 02 Thread =      - - - - -
Pattern 01 Thread =      + + + + +
Pattern 02 Thread =     - - - - - -
Pattern 01 Thread =     + + + + + +
Pattern 02 Thread =    - - - - - - -
Pattern 01 Thread =    + + + + + + +
Pattern 02 Thread =   - - - - - - - -
Pattern 01 Thread =   + + + + + + + +
```

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 for meal preparation of a Restaurant. You can prepare three meals for the day (**Breakfast, Dinner, and Lunch**) with time duration of (**60 minutes, 45 minutes,** and **30 minutes** for each).

i). Implement two interfaces **IPrepareQuickly** and **IPrepareDeliciously**. Each interface you should declare methods (in **IPrepareQuickly** interface declare the method **void deliveryTime()** and in **IPrepareDeliciously** interface declare methods **void addFlavour()** and **double getCost())**

(02 marks)

ii). Then create 3 classes **ChickenFlavour**, **FishFlavour**, and **EggFlavour** and those classes should implement the **IPrepareDeliciously** interface and override all methods with in the class.

(06 marks)

iii). Similarly create another 3 classes **OneHour**, **ThirtyMinutes**, and **FourtyFiveMinutes** and those classes should implement the **IPrepareQuickly** interface and override the method as well.

(03 marks)

iv). Create an **Abstract** class called **Meal** and aggregate two interfaces **(IPrepareQuickly,** and **IPrepareDeliciously),** you should set those two behaviors with using two set methods **setFlavour()** and **setDuration()**. (Those "set" methods are used to dynamically add prepare **quickly** and **prepare deliciously** features to meal)

(06 marks)

b) Now for the above three meals you can add different flavors such as **chicken, fish**, and **egg** and the cost of each flavored meal respectively chicken - 100/=, Fish – 80/=, and egg – 60/= rupees. Based on the flavor cost should be different.
[Assumption: You can't add more than one flavor per meal]

i). Then implement another two methods called **mealWithFlavour(),** and **mealInDuration()** and you should call relevant **addFlavour()** and **deliveryTime()** method respectively through the declared interfaces of the Meal class

(02 marks)

ii). Apart from that with in the **Meal** class you should add two **abstract** methods **displayMeal() and displayCost()**

(01 mark)

iii). Now **extends** the **Meal** class in the **Breakfast, Lunch and Dinner** classes. Implement all abstract methods. Within the **displayMeal()** method you should call for the **mealWithFlavour(), mealInDuration(), and displayCost() methods.**

(10 marks)

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

Save the project as **Question04**

Sample Output:

ThirtyMinutes.java | EggFlavour.java | Breakfast.java | Meal.java

```
1  package oop.design.patterns;
2
3  public class TestStratergy {
4
5      public static void main(String[] args) {
6
7          Meal meal = new Breakfast();
8          meal.setFlavour(new ChickenFlavour());
9          meal.setDuration(new ThirtyMinutes());
0          meal.displayMeal();
1
2          Meal meal2 = new Lunch();
3          meal2.setFlavour(new FishFlavour());
4          meal2.setDuration(new OneHour());
5          meal2.displayMeal();
6
7          Meal meal3 = new Dinner();
8          meal3.setFlavour(new EggFlavour());
9          meal3.setDuration(new FourtyFiveMinutes());
0          meal3.displayMeal();
1
2      }
3
4  }
5
```

Console | Problems

```
<terminated> TestStratergy [Java App

Preparing Breakfast......
Added Chicken for the meal
Meal is ready in 30 minutes
Cost for the meal is = 100.0

Preparing Lunch....
Added fish for the meal
Meal is ready in 60 minutes
Cost for the meal is = 80.0

Preparing Dinner.......
Added egg for the meal
Meal is ready in 45 minutes
Cost for the meal is = 60.0
```