**Centennial College**

**COMP 228: Java Programming**

**LAB #3 – Using Inheritance and Polymorphism**

**Studen**t: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Due Date: November 13th, 2023

Purpose: The purpose of this Lab assignment is to:

1. Practice the use of Inheritance
2. Practice the use of Polymorphism.

References: Learning materials for weeks 5 and 6, textbook, and other references (if any)

This material provides the necessary information you need to complete the exercises.

Be sure to read the following general instructions carefully:

This lab should be completed individually by all the students. You will have to demonstrate your solution in a scheduled lab session and submitting the project **through the dropbox link on D2L**.

You must name your IntelliJ project according to the following rule:

**YourFullName\_COMP228Labnumber**

Example: **JohSmith\_COMP228Lab3**

Each exercise should be placed in a separate package named *exercise1*, *exercise2*, etc.

Submit your assignment in a **zip file** that is named according to the following rule:

**YourLastName\_COMP228Labnumber.zip**

Example: **JohSmith\_COMP228Lab3.zip**

Apply the naming conventions for variables, methods, classes, and packages:

- *variable names* start with a *lowercase* character

- *classes* start with an *uppercase* character

- **packages** use only *lowercase* characters

- *methods* start with a *lowercase* character

## Exercise 1

Write a Java application that implements different types of insurance policies for employees of an organization.

Let **Insurance** be an abstract superclass and **Health** and **Life** two of its subclasses that describe respectively health insurance and life insurance.

The **Insurance** class defines an instance variable of type **String** to describe the **type of insurance** and an instance variable of type **double** to hold the **monthly cost** of that insurance.

Implement the **get** methods for both variables of class **Insurance**. Declare also two **abstract** methods named **setInsuranceCost()** and **displayInfo()** for this class.

The **Life** and **Health** class should implement **setInsuranceCost** and **display** methods by setting the appropriate monthly fee and display the information for each insurance type.

Write a driver class to test this hierarchy. This application should ask the user to enter the type of insurance and its monthly fee. Then, will create the appropriate object (Life or Health) and display the insurance information.

As you create each insurance object, place an **Insurance** reference to each new **Insurance** object into an array. Each class has its own **setInsuranceCost** method. Write **a polymorphic** screen manager that walks through the array sending **setInsuranceCost** messages to each object in the array and displaying this information on the screen.

(50 marks)

**Exercise #2:**

Create an interface called GameTester. The GameTester interface includes a name for the game tester and declare an abstract method to generate paystub. Implement interface into both subclasses.

Create two subclasses called FullTimeGameTester, PartTimeGameTester. Create a console application that demonstrates how to create objects of both subclasses. Ask user to enter type of game tester, salary information and generate paystub according to game tester.

(50 marks)

**Evaluation:**

|  |  |
| --- | --- |
| **Functionality** |  |
| Correct implementation of classes (instance variable declarations, constructors, getter and setter methods, etc.) | 40% |
| Correct implementation of driver classes (declaring and creating objects, calling their methods, interacting with user, displaying results) | 40% |
| Comments, correct naming of variables, methods, classes, etc. | 5% |
| **Friendly input/output** | 15% |
| **Total** | 100% |