OBJECT ORIENTED PROGRAMMING (JAVA) (CST8284)

LAB 7

Implementing Interfaces

Important Instruction

- Carefully review **all files** provided to you to understand the logic and hierarchy involved in this application, and what you need to do.
- Do **NOT** proceed with the tasks in this slide deck until you have reviewed each code file to ensure that you are doing the right thing.
- Code files required to work on this Lab has been provided (except the one you will create). Pay attention to comments inserted in the code files.
- Note that "- use" is NOT a part of the file name. Save the file with the name provided but without the **"– use"** part.

Overview

- In this lab you will leverage on your knowledge of **abstract** classes, **inheritance**, **polymorphism** and **interfaces** to implement polymorphic behavior on an interface.

- You are required to modify the account **Payme** application that has been provided to you in this lab, such that it will include the full functions of the application you worked on in your Lab 6.

- The **modified application** should be able to process two **invoice objects**, and now will also process each object of the **Programmer** subclasses.

Overview (2)

- If the object currently being processed is a **BasePlusCommissionProgrammer**, then the application should **increase** the BasePlusCommissionProgrammer's base salary by **10%**.
- Your **output file** should show the **payment amount** for each object processed.

PROVIDED FOR YOU…

Items provided for you include:

- **Java code files and sample output are** provided for you in this lab. You will need to create, update and/or modify specific files. Items provided include:
  - **Programmer.java** (**abstract** Superclass that **implements** **Payme** interface)
  - **Invoice.java** (unrelated class that **implements** **Payme)**
  - **PaymeInterfaceTest.java** (provides the **main method** to test your classes)
  - Sample **output** file (just a sample)

Items you will reuse from Lab 6

- Review the following **Java code files** provided for you that you will reuse in this lab.

  - **CommissionProgrammer.java** (programmers who are paid based on commission. Extends **Programmer**)
  - **HourlyProgrammer.java** (programmers who are paid per hour. Extends **Programmer**)
  - **SalariedProgrammer.java** (regular salaried programmers. Extends **Programmer**)
  - **BasePlusCommissionedProgrammer** (extends **CommissionedProgrammer**)

YOUR TASKS…

Your Tasks: Declare the Payme interface

- **Payme.java interface declaration**
  - Create a **Payme.java** file (this is the Payme interface declaration).
  - Should contain a method of type **double** called **getPaymentAmount()** for calculating payment (but **no implementation**).
- It is important to note that both **Invoice** and **Programmer** implements interface **Payme.**

Your Tasks…(2)

- **Review the** code file for **Programmer** provided for you, and include required code to **all** the **sections** that have been marked.

- **Do not** modify the **Invoice** code file.

Your Tasks…(3)

- **Modify** the following codes files that were given to you named: **HourlyProgrammer**, **SalariedProgrammer**, and **CommissionProgrammer** so as to place them in the **Payme** hierarchy (as subclasses of the **Programmer** code file provided for you). Note that "-use" is not a part of the class name.

  - **Hint:** Look through the code in the **these subclasses** and **implement** method **getPaymentAmount** in each class.
  - **Explain** the reason why you had to do so to your professor. **Do you know why?**

ALGONQUIN COLLEGE

Your Tasks…(4)

- **Modify** the **BasePlusCommissionProgrammer** so that it **extends** the current version of the **CommissionProgrammer** class that you created in your Task 1 (previous slide).

- **Modify** the launcher (test) file for this application provided to you, called the **PaymeInterfaceTest**, so that it can process two **Invoices** polymorphically, one HourlyProgrammer, one CommissionProgrammer, one SalariedProgrammer, and one BasePlusCommissionProgrammer.

Your Tasks… (5)

For your output file be sure to:

- Output a String representation of each **Payme** object.

- **show in your code the fact that**: If an object is a **BasePlusCommissionProgrammer,** increase the base salary by **10%** and show this in your output.

- Show the output of the **payment amount** for each **Payme** object.

- See the sample output format provided. Reuse the details provided in the output file (such as first name, last name, etc. but use your name as the last name listed). Remember that this is just a sample file.

DEMO YOUR LAB…

Demo your lab to the Professor (RUBRICS)

- Show your code for **Payme** interface declaration
- Show your professor the updates you made to the classes: **Programmer, HourlyProgrammer**, **CommissionProgrammer**, **BasePlusCommissionProgrammer, SalariedProgrammer** and **PaymeInterfaceTest.java**
- Run your code to show that it works correctly
- Show your output file to your Professor
- Prepare to answer some questions.