ALGONQUIN COLLEGE

OBJECT ORIENTED PROGRAMMING (JAVA) (CST8284)

LAB 6: Important Information

❖ 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.

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 an application.

❖ The **modified application is required** to process two **invoice objects**, and will also process each object of the subclasses of a super class called **Programmer**. **Invoice** and **Programmer** are unrelated classes.

ALGONQUIN COLLEGE

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.

Items provided for you include:

❖ **Java code files and sample output are** provided for you in this lab. You will need to <u>create</u>, <u>update</u> and/or <u>modify</u> 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)

ALGONQUIN COLLEGE

Items provided for you (2)

➤ **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**)

➤ **BasePlusCommissionProgrammer** (extends **CommissionProgrammer**)

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.

❖ It is important to note that both **Invoice** and **Programmer** implements interface **Payme.**

ALGONQUIN COLLEGE

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.

ALGONQUIN
COLLEGE

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).

➢ **Hint:** Look through the code in the **these subclasses** and change the name of the **earning** method to **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 <u>test file</u> for this application provided to you, called the **PaymeInterfaceTest**, so that it can process two **Invoices** <u>polymorphically</u>, one HourlyProgrammer, one CommissionProgrammer, one SalariedProgrammer, and one BasePlusCommissionProgrammer.

ALGONQUIN COLLEGE

Your Tasks (5)
Modify the Test File (2)

❖ Create an array of Programmer variables to store the references to objects of each concrete class in your Programmer hierarchy.

❖ This should show the String representation and each of the existing Programmer payments (their earnings).

Your Tasks… (6)

For your output file be sure to:

➢ <u>Output</u> 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 <u>output</u> 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. <u>but use your name </u>as the last name listed). Remember that this is just a sample file.

ALGONQUIN COLLEGE

Part B: To Be Announced in 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 <u>correctly</u>

❖ Show your output file to your Professor

❖ Prepare to answer some questions.

ALGONQUIN
COLLEGE