UNIVERSITY OF TECHNOLOGY, JAMAICA

SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

ADVANCED PROGRAMMING (CIT3009)

TUTORIAL ONE – Abstract Classes, Interface, Inheritance, Polymorphism & Cast

DATE GIVEN: Week one

OBJECTIVES: The purpose of this tutorial, is give students a refresher walk through of the concepts of Abstract Classes, Interfaces, Inheritance, Polymorphism and Cast, covered in Lecture one.

INSTRUCTION: Students are required to complete all exercise listed below during the class session. To facilitate a desirable outcome, students are encouraged to use the names given to classes, interfaces, variables, etc. in the tutorial document.

Exercise one

1. Create a new Java Project named "APTutorialOne"
2. Create a package named domain.
3. In the domain package, create an abstract class named Employee. The Employee class should have five (5) attributes: ID (String), name (String), phone (String), salary (double) and hoursWorked (double). Create the constructors, accessors, mutators and toString methods for the class.
4. In the Employee class, create an abstract method named calculatePay. The method takes no parameter and returns a double value.

Exercise two

5. In the domain package, create an interface and name it Payable. In the Payable interface, declare an identifier of datatype double, named OVTRATE and assign it a value of 1.25
6. Create an abstract method named calculateOverTime, which takes no arguments and returns a double value.

Exercise three

7. In the domain package, create a class named "PartTimeEmployee", which inherits from the Employee class and implements the Payable interface.
8. Implement the abstract calculatePay method, and the abstract calculateOverTime method inherited from the Employee class and Payable interface, respectively.
9. In the PartTimeEmployee class, create two attributes: BASICPAYRATE, which should be assigned a constant value of 3575.65; and overTimePay of type double.

10. Create the default constructor, primary constructor and another constructor that takes an Employee's ID, name, phone number and hours worked as arguments.
11. In the calculatePay method, implement the code depicted in the following image:

```
if (hoursWorked <= 6) {
     return salary = BASICPAYRATE * hoursWorked;
}

return salary = BASICPAYRATE * 6 + calculateOverTime();
```

12. In the calculateOverTime method, implement the code depicted in the following image:

```
overTimePay = (BASICPAYRATE * (hoursWorked - 6) * OVTRATE);
return overTimePay;
```

Exercise four

13. Create a package named driver. In the driver package, create a class named Driver, which contains the main method.
14. In the main method, create an object of the Payable interface and assign it the reference of a PartTimeEmployee using the constructor that accepts an ID number, name, phone number and hours worked, as arguments.
15. Declare a local variable named overtime. Using the object of the Payable interface, from 14 above, call the calculateOvertime method, and assign the returned value to the variable "overTime". You may wish to pass the overtime variable to a print function to display the overtime earned.
16. Create an object of the Employee class and assign to it, the reference of the Payable object from 14. You should get an error when assigning the reference of the Payable object to the Employee object. To correct this error, cast the Payable object to an Employee. Ex.
    ```
    Employee emp = (Employee)taxPayer;
    ```
17. Declare a local variable named salary. Using the object of the Employee class, from 16 above, call the calculatePay method, and assign the returned value to the variable "salary". You may wish to pass the salary variable to a print function to display the salary earned.
18. Pass the Payable object and the Employee object, to print functions to display their data. Your main method should look similar to the following:

```
public static void main(String[] args) {

    Payable taxPayer = new PartTimeEmployee("1605308", "Johnny Public", "1(876)378-7659", 10.5);
    double overTime = taxPayer.calculateOverTime();

    System.out.println("Over-time pay: " + overTime + "\n");
    System.out.println(taxPayer);

    Employee emp = (Employee)taxPayer;
    double salary = emp.calculatePay();

    System.out.println("Gross Salary: " + salary);

    System.out.println(emp);
}
```