DePaul University
College of Computing and Digital Media

CSC 211 - Programming in Java I

Assignment 5

In this assignment, we're going do the following:

- Add one new data attribute to the Employee class - a Timecard – so that the Employee will now *own* a Timecard object.
- Update the Employee's constructor code so that it will create it's own Timecard object
- Add a "print" method to the Employee & Timecard
- Add a "getPay()" method to the Employee class

1) You can re-use your project from Assignment #4. There is no need to create a new project for this assignment. To do this, make a copy of your project folder from Assignment 4, and give the copy a new name for Assignment 5. For example, my Assignment 4 was done in "C:\Documents and Settings\hieldc\Desktop\Assignment4". I made a copy of the Assignment 4 folder and named it "Assignment5" as follows:

   "C:\Documents and Settings\hieldc\Desktop\Assignment5".

2) Now open the copied project in NetBeans. We need to rename the copied project within NetBeans. To do this, right-click on the existing "Assignment4" project and select menu option "Rename…". Change the "Project Name" from "Assignment4" to"Assignment5" and click "Rename".

3) Finally, right-click on the "Assignment5" project and select menu option "Set as Main Project".

4) First, you must add a "`public String toString()`" method to your Timecard class. The resulting String should look just like this:

```
Weekly Hours: 38
       Day 1: 8
       Day 2: 7
       Day 3: 9
       Day 4: 8
       Day 5: 6
```

5) Next, we need to add a new data attribute to your *Employee (the Employee needs to own a Timecard data attribute)* – the type of the data member is **Timecard**, call it what you want. (I called mine "timecard" – creative, eh?)

6) Create a *private* accessor and private modifier (get & set) for the new Timecard data attribute. *(We don't want any other object to have access to our Timecard directly so we make the get & set private).* The only error checking you need to do in the modifier is to verify that the Timecard parameter is not null. If it is null, that's bad – handle that situation like we handle other "set" method errors.

7) Add an *additional* **"int" array** parameter to the Employee **constructor**, at the end of the current parameter list. This "int" array is the array of hours per days needed to create a new Timecard object from within the Employee constructor (I called "int" array parameter "daysIn").

   Whenever you want to declare a method or constructor with an ***array*** *as one of the parameters, you do it like this:*

   Method: type methodName(…,…,…, type[] variableNameIn);
   Constructor: ConstructorName(…,…,…, type[] variableNameIn);

   Example:
   Sample Method: public void doSomething(…,…,…, int[] myIntsIn);
   Sample Constructor: public WorkSchedule(…,…,…, int[] myIntsIn);

> ***Info**: Since the Employee **owns** the reference to a Timecard object, and we want no other part of the application to have a reference to it’s content, it is a good practice to have the Employee object **create** the Timecard object **itself** in the Employee’s constructor rather than create the Timecard in the “main” like we did in Assignment 5.*
>
> *Recall that the Timecard constructor needs an “int” array. This just means that we need an “int” array in order to create a Timecard. So, that’s why we’ll pass the Employee constructor the new “int” array parameter – so that it can use that array when it creates a new Timecard object and calls it’s constructor.*

Now, add one new line to the end of the Employee constructor – a call the Timecard data member’s “set” method, passing it a *new Timecard object*. The new Timecard object is created by passing the “int” array parameter “daysIn” (or whatever you called it) to the Timecard’s constructor.

Similar Examples:

```
setAnimal( new Animal(nameIn) );
setWorkSchedule( new WorkSchedule (timeIn) );
```

8) Now, we will add one new functional method to the Employee class called “getWeeklyPay()”. All this method needs to do is return the result of the following:

   Call the “getWeeklyHours()” method of the Employee’s timecard attribute and multiply the returned value by the Employee’s hourlyRate attribute.

9) Finally, you must add a “public String toString()” method to your Employee class. The resulting String should look just like the below *(the Employee’s “toString” should make use of it’s Timecard’ data member’s “toString” method to generate the weekly hours and daily hours info):*

```
Name:           George Orwell
Id:             1984
Hourly Rate:    $19.99
Weekly Hours:   42
      Day 1: 9
      Day 2: 8
      Day 3: 10
      Day 4: 8
      Day 5: 7
Weekly Pay:     $839.58
```

10) Now – change the “main” method of your “Driver” class to look like the following *(you can completely replace the existing “main” method in Driver with this one):*

```
public static void main(String[] args)
{
    // Get input data values from the user (via the keyboard)
    Scanner userInput = new Scanner(System.in);

    // Now declare local variables to hold user inputs
    // needed by the Employee constructor
    String firstNameInput;
    String lastNameInput;
    int idInput;
    double rateInput;

    // Create a local int array with the same number of
    // elements as the Timecard's "numDays" attrbiute
    int[] workedDays = new int[Timecard.numDays];

    // Get the  values needed by Employee
```

```
        // constructor from the user (via the keyboard)
        System.out.println("Employee First Name: ");
        firstNameInput = userInput.next();

        System.out.println("Employee Last Name: ");
        lastNameInput = userInput.next();

        System.out.println("Employee Id: " );
        idInput = userInput.nextInt();

        System.out.println("Employee Hourly Rate: ");
        rateInput = userInput.nextDouble();

        // Now, in a "for" loop, query the user for each days hours and
        // store the response in the individual int array elements
        for (int i = 0; i < Timecard.numDays; i++)
        {
            System.out.println("Enter Hours for day " + (i + 1) + ": ");
            workedDays[i] = userInput.nextInt();
        }

        // Now allocate a new instance of an Employee, passing the test
        // data (and the int array we created above) to the constructor.
        Employee e =
               new Employee(firstNameInput, lastNameInput, idInput, rateInput, workedDays);

        // Print the Employee - this will invoke the Employee's "toString()" method.
        System.out.println("\nEmployee:\n-------------------------------------");
        System.out.println(e);
    }
```

11) Now compile your project – the “Driver.java”, “Employee.java” & “Timecard.java” files will be compiled. Fix any compiler errors as usual. Then - run the program.

12) Done! *(Be SURE to test your program with good AND bad values to be sure your error checking is working properly!)*

13) Example Inputs & Outputs

```
Employee First Name:
Rachel
Employee Last Name:
Weisz
Employee Id:
9724
Employee Hourly Rate:
23.45
Enter Hours for day 1:
8
Enter Hours for day 2:
9
Enter Hours for day 3:
4
Enter Hours for day 4:
7
Enter Hours for day 5:
8

Employee:
------------------------------------
Name:           Rachel Weisz
Id:             9724
Hourly Rate:    $23.45
Weekly Hours:   36
        Day 1: 8
        Day 2: 9
        Day 3: 4
        Day 4: 7
        Day 5: 8
Weekly Pay:     $844.20
```

Submission:

- This assignment is due before the start of class next week (on or before 5:45 pm on Monday, May 17th). Late assignments will be penalized 10% per week.
- Your submission should consist of your entire Assignment 5 project folder put into a single ZIP file (or a “TAR” file, or a “RAR” file). Check with me on other formats.
- All submissions are to be made via the course’s Course OnLine site
- You may email me with any questions on this assignment at any time between now and the due date.