DePaul University
College of Computing and Digital Media

CSC 211 - Programming in Java I

Assignment 4

Your assignment this time is to add a new class called "Timecard" (we will use this Timecard class with your Employee class in future assignments):

1) You can re-use your project from Assignment #3. There is no need to create a new project for this assignment. To do this, make a copy of your project folder from Assignment 3, and give the copy a new name for Assignment 4. For example, my Assignment 3 was done in "C:\Documents and Settings\hieldc\Desktop\Assignment3". I made a copy of the Assignment 3 folder and named it "Assignment4" as follows:

   "C:\Documents and Settings\hieldc\Desktop\Assignment4".

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 "Assignment3" project and select menu option "Rename…". Change the "Project Name" from "Assignment3" to"Assignment4" and click "Rename".

3) Finally, right-click on the "Assignment3" project and select menu option "Set as Main Project".

4) Create a new Java class called "Timecard" in the existing "coursework" package. We've created new classes before so you should know how to do this by now. If you have forgotten, refer to the NetBeans notes pack, or the instructions in your previous assignments.

5) Now, add the following data members to your Timecard:

   - A public "static final int" variable "numDays" to hold the number of days in a work-week – initialize this to 5. *(This **must** be declared & initialized **before** you declare the next data attribute)*

   - A private "int" *array* created as size "numDays". The individual "slots" in this array will represent days of the work-week. The "int" we store in each slot will indicate the hours worked in that day). I'll refer to this new int array as "daysOfTheWeek" in this handout.

Example: "daysOfTheWeek" :

Slot [0]	*8*
Slot [1]	*5*
Slot [2]	*6*
Slot [3]	*3*
Slot [4]	*9*

5

6) Next, we'll create the accessor & modifier for the "daysOfTheWeek" data attribute as described below. Once they are done, we'll tackle the constructor.

   - **NOTE**: Rather than try to "set" or "get" the whole "int" array at once with the usual kind of "get" & "set" methods, we'll instead create a slightly different kind of get & set method that will get & set only *one* element of that array at a time: *(so don't create a "get" & "set" that returns the whole "int" array)*

     a. Create a public "int" method called "getHoursByDay" that takes one "int" parameter for the day whose hours you want to get. This should return a single "int" from the "daysOfTheWeek" array using the "day" parameter as the index into the array.

NOTE:

- *The "day" parameter passed in must be greater than or equal to 0 and less than the "numDays" you created earlier. Otherwise,this is an error - you will be accessing space beyond your "daysOfTheWeek" array. If this happens you should print an error message to the screen and exit the program.*

b. Create a "void" method called "setHoursByDay" that takes 2 parameters:
   - an "int" parameter for the day whose hours you want to set
   - an "int" parameter for hours value you want to set for the day in question

   This method should set one element of the "daysOfTheWeek" array using the "day" parameter as the index into the array and the "hours" parameter as the value to set to.

   NOTE:
   - *The "day" must be greater than or equal to 0 and less than the "numDays" you created earlier. Otherwise, you will be accessing space beyond your "daysOfTheWeek" array.*
   - *The "hours" value must be greater than or equal to 0 and less than or equal to 24.*
   - *If either values passed in are 'bad", you should print an error message to the screen and exit the program.*

7) Now we will create a constructor that uses the "set" method we just created. Create a constructor that accepts 1 parameter - an "int" array *(int[])* that we will pass in for the hours worked for each day of the work-week. *You should immediately check to see if the int array passed in is null. If it is, you should print an error message to the screen and exit the program.*

This constructor needs to copy the individual "int" values from the array parameter passed in into your "int" array data attribute "daysOfTheWeek".

To do this, you should use a "for" loop to copy the individual "int" values from each index (slot) of the array parameter passed in to the "int" slots of your "daysOfTheWeek" data attribute. You can use the "setHoursByDay" method you previously wrote to assist with this.

For example: *(this is sample pseudocode – not real Java code – do not type this in!):*

```
for (int i = 0; i < the array's size; i++)
       setHoursByDay(i, theParameterArray[i]);
```

That's all the constructor needs to do!

8) Now, we'll create a functional method called "getWeeklyHours()" that will calculate & return the (int) total hours worked for the week that the Timecard represents.

This is calculated by summing up all the int's in the "daysOfTheWeek" array and return that value:

- Create a temporary "int" variable that we will use to hold the sum of the hours of all the days in the "daysOfTheWeek" array data attribute. I called mine "count".
- Now, for each "int" in the "daysOfTheWeek" array (use a "for" loop!) add that "int" to that "count" variable we just created.
- Finally – return the "count" variable.

9) Now – make the following changes to the end of your existing "Driver.java" file. We're going to add a little more code to the current "main" method there. Just BEFORE the close-brace of the existing "main" method, add the following:

```
    […] // Existing "main" code remains the same

    // Here create an "int" array to pass to the Timecard constructor
    int[] workedDays = new int[Timecard.numDays];


    // Now read keyboard input for each of the array slots.
    for (int i = 0; i < Timecard.numDays; i++)
    {
        System.out.println("Enter Hours for day " + (i + 1) + ": ");
        workedDays[i] = userInput.nextInt();
    }


    // Create the Timecd using the newly-filled array
    Timecard t = new Timecard(workedDays);


    // Print the daily hours values from the Timecard to be sure they're as expected
    for (int i = 0; i < Timecard.numDays; i++)
    {
        System.out.println("Value of hours for day " + (i + 1) + ": " + t.getHoursByDay(i));
    }


    // Now print the total of the weekly hours to make sure our method works properly
    System.out.println("Total Weekly Hours: " + t.getWeeklyHours());


} // Existing close-brace for "main" method
```

10) Now compile your project – the "Main.java", "Employee.java" & "Timecard.java" files will be compiled. Fix any compiler errors as usual. Then - run the program.

11) Done! Example Inputs & Outputs:

(You need to type in the values where shown below in blue. You can use your own data for these values)

(Run 1 – using no bad data)

```
Employee First Name:
Victoria
Employee Last Name:
Beckham
Employee Id:
8924
Employee Hourly Rate:
14.99

Name: Victoria Beckham
Emp. Id: 8924
Hourly Rate: $14.99

Enter Hours for day 1:
8
Enter Hours for day 2:
5
Enter Hours for day 3:
7
Enter Hours for day 4:
9
Enter Hours for day 5:
8
Value of hours for day 1: 8
Value of hours for day 2: 5
Value of hours for day 3: 7
Value of hours for day 4: 9
Value of hours for day 5: 8
Total Weekly Hours: 37
```

(Run 2 – using a bad hours value for day 5)

```
Employee First Name:
```

```
Mel
Employee Last Name:
Gibson
Employee Id:
2830
Employee Hourly Rate:
12.49

Name: Mel Gibson
Emp. Id: 2830
Hourly Rate: $12.49

Enter Hours for day 1:
2
Enter Hours for day 2:
4
Enter Hours for day 3:
8
Enter Hours for day 4:
16
Enter Hours for day 5:
32
Bad hours value used in setHoursByDay: 32
```

(Run 3 using a bad hours value for day 1)

```
Employee First Name:
Catherine
Employee Last Name:
Zeta-Jones
Employee Id:
6549
Employee Hourly Rate:
16.09

Name: Catherine Zeta-Jones
Emp. Id: 6549
Hourly Rate: $16.09

Enter Hours for day 1:
-2
Enter Hours for day 2:
8
Enter Hours for day 3:
3
Enter Hours for day 4:
14
Enter Hours for day 5:
9
Bad hours value used in setHoursByDay: -2
```

Submission:

- This assignment is due before the start of class next week (on or before 5:45 pm on Monday, May 10th). Late assignments will be penalized 10% per week.
- Your submission should consist of your entire Assignment 3 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.