

## CS520 Module 2 Assignment

### General Rules for Homework Assignments

- You are strongly encouraged to add comments throughout the program. Doing so will help your facilitator to understand your programming logic and grade you more accurately.
- You must work on your assignments individually. You are **not allowed** to copy the answers from the others. *However*, you are encouraged to discuss approaches to the homework assignment with your section mates and the facilitator in your section via the discussion board.
- Each assignment has a strict deadline. However, you are still allowed to submit your assignment within 2 days after the deadline with a penalty. 15% of the credit will be deducted unless you made previous arrangements with your facilitator and professor. Assignments submitted 2 days after the deadline will not be graded.
- The assignment solution will be available 48 hours after the assignment deadline.
- When the term *lastName* is referenced in an assignment, please replace it with your last name.

**You are strongly encouraged to add comments into your program!**

Create a new Java Project in Eclipse named HW2\_*lastName* and complete the following two parts.

## Part1 (50 Points)

Create a package named *cs520.hw2.part1* and complete the following by writing the appropriate classes under this package.

1. The course tuition is equal to the “credits for the course” multiplied by “the cost per credit.” Create a “procedural style” program that meets the requirements below. Create a class named *TuitionProceduralStyle* and test the tuition of two different courses as specified below.

- a. Prompt the user to enter the following values for the first course one by one through a dialog box.
  - The *name* of the first course, the *number of credits* for the first course (integer type), and the *cost per credit* for the first course (integer type)
- b. Now, prompt the user to enter the following values for the second course one by one through a dialog box.
  - The *name* of the second course, the *number of credits* for the second course (integer type), and the *cost per credit* for the second course (integer type)
- c. Compute the course tuition for the first course and store it into an appropriate variable type.
- d. Display the first course’s name and its tuition using a dialog box.
- e. Compute the course tuition for the second course and store it into an appropriate variable type.
- f. Display the second course’s name and its tuition using a dialog box.
- g. Using the *if-else* construct, check which tuition is greater than the other and display the information and the difference using a dialog box. Also, check if the tuitions are the same as part of this comparison.
- h. Make sure to test your program with values for all the three scenarios possible in the above step.

2. Transform the procedural style computation of the previous problem into an object-oriented application. In order to do this, create two classes named *Course* and *TuitionObjectStyle* as specified below.

- a. The class named *Course* describes each course and should have
  - The instance (or member) variables – *courseName* (*String*), *costPerCredit* (integer), and *numberOfCredits* (integer).
  - The appropriate *set* methods for the above three variables.
  - The appropriate *get* methods for the above three variables.
  - The method *getTotalTuition* which returns the tuition for this course.
  - The method *printTuitionDetails* which displays the name and the tuition of this course using a dialog box.
- b. The class named *TuitionObjectStyle* creates two course objects and tests them as follows:
  - In the *main* method, instantiate two objects of the type *Course* named *course1* and *course2*.
  - Perform the equivalent steps to the ones carried out in Step 1.

## Part2 (50 Points)

Create a package named *cs520.hw2.part2*. Using this package, create the following classes and implement the specified functionality.

1. Create a class named *Course* as follows.
  - a. The instance (or member) private variables – *courseName* (String), *costPerCredit* (integer), and *numberOfCredits* (integer).
  - b. Only a single constructor with three arguments, the name, the cost per credit, and the number of credits. Set the instance variables with the specified argument values. Output to the console the values using the *get* methods.
  - c. The public *set* and *get* methods for the three instance variables.
  - d. A public *getTotalTuition* method that returns the tuition for the course.
  - e. Override the *toString* method to return the string representation of this object in the format "Course:<course name> @ \$<total tuition>".
2. Create a derived class from the *Course* class named *OnlineCourse*.
  - a. Only one private instance variable – *technologyFee* (integer) – is allowed in the *OnlineCourse* class.
  - b. Only a single constructor with four arguments, the name, the cost per credit, the number of credits, and the technology fee. Output to the console the values using the *get* methods. The technology fee specified is per credit for the online class.
  - c. The public *set* and *get* method for the technology fee only.
  - d. Override the *getTotalTuition* method which returns the sum of the technology fee and the tuition as computed in the *Course* class. **No assumption should be made on how the tuition is computed by the *Course* class.**
  - e. Override the *toString* method to return the string representation of this object in the format "OnlineCourse:<course name> @ \$<total tuition>".
3. Create a *Test* class to test the following functionality in its *main* method.
  - a. Declare a variable named *currentCourse* of type *Course*.
  - b. Create an *Course* object with appropriate values and assign it to the above variable.
  - c. Print to the console the string representation of this object.
  - d. Create a *OnlineCourse* object with appropriate values. Reassign the *currentCourse* to this object.
  - e. Print to the console the string representation of this object.

The sample output for Part2 is shown below.

Taking a regular course...

In Course Constructor values set:

CourseName = CS520, CostPerCredit = \$390, NumberOfCredits = 4

Printing...

Course:CS520 @ \$1560

Taking an online course...

In Course Constructor values set:

CourseName = CS520, CostPerCredit = \$760, NumberOfCredits = 4

In OnlineCourse Constructor values set:

CourseName = CS520, CostPerCredit = \$760, NumberOfCredits = 4, TechnologyFee = \$60

Printing...

OnlineCourse:CS520 @ \$3280

## Submission

Create an archive of your Eclipse project using the following steps. Select the HW2\_*lastName* project in the Eclipse IDE's *Package Explorer* or the *Navigator* window.

Click *File->Export*. Select the *General->Archive File* option. Click *Next*.

Specify the "To archive file:" entry as say, C:\Temp\HW2\_*lastName*.zip.

The zip file will be created and stored in the C:\Temp folder.

Submit this zip file as an attachment in the Assignment Section.