CIS280 Project #4 (Chapter 6)

Name(s):_____ Due: _____

Personalization and Commenting Requirements:

- ***All source code must contain a comment with the source file name, programmer(s) names(s) and date created***
- ***Commenting at top should include the purpose of the program***
- ***Appropriate, clear commenting must be used throughout the code; be professional and pay attention to spelling***
- ***All applications should start with an output to the screen that contains the name of the application and the programmer(s***)
- ***Name the programs as shown on this project sheet* Chapter 6 Requirements**

1. *TotalFees.java* - This application will demonstrate *overloading methods.* Write an application that includes the following variables: tuitionFee, bookFee, techFee, and parkingFee, all of which are double variables. You may simply initialize these variables with values that you make up rather than gather keyboard input. You may choose either GUI or console-based. All of this can be done in one program that includes a main method. (In other words it does not require that a class definition and driver be code separately.)

   Create four overloaded computeTotal() methods:
   a. When computeTotal() receives tuitionFee only, display console-based output that says "Tuition only" along with the tuitionFee.

   b. When computeTotal() receives both tuitionFee and bookFee, display "Tuition and Books" along with the total of the two variables.

   c. When computeTotal() receives tuitionFee, bookFee, and techFee display "Tuition, Books and Technology" along with the total of all three variables.

   d. When computeTotal() receives all of the variables, display "The Whole Package" along with the total of all the variables

2. *YourChoice2.java* and *DemoYourChoice2.java* - Modifications of an earlier assignment
   a. *YourChoice2.java* --- Copy *YourChoice.java* from Project 2 into *YourChoice2.java*
      i. Add a static variable to your class
      ii. Add a constructor that receives parameters to your class (if you don't already have one)
      iii. Add a *toString* method to your class that displays the data in your object in a nice format

   b. *DemoYourChoice2.java* --- Copy *DemoYourChoice.java* from Project 2 into *DemoYourChoice2.java*
      i. Add code to instantiate an <u>additional</u> object using the constructor that receives parameters
      ii. Add code to display data from the newly added object by calling the *toString* method you added in the class
      iii. iii. Demonstrate use of the static variable

3. *Classification.java* – using enumerated data type
   Write an application that will tell a student what year in college he/she is in. You may choose either GUI or console-based. All of this can be done in one program that includes a main method. (In other words it does not require that a class definition and driver be code separately.)
    a. Use an enumerated data type to set up a data type called *collegeYear* that contains the predefined values FRESHMAN, SOPHOMORE, JUNIOR, or SENIOR
    b. Ask the user for the number of credits he/she has completed.
    c. Determine and display collegeYear based on:
        a. 0 – 30 credits is a freshman
        b. 31 – 60 credits is a sophomore
        c. 61 – 90 credits is a junior
        d. 91+ credits is a senior

Submission Requirements

- Zip your .java source files into a folder labeled with your name(s) and project number. Submit this zipped folder in Blackboard. Please use the onboard zip utility in Windows rather than a third-party application.