



DOUGLAS COLLEGE

COMMERCE AND BUSINESS ADMINISTRATION

CSIS 2175: ADVANCED INTEGRATED SOFTWARE DEVELOPMENT

ASSIGNMENT 1

Due date: February 8, 2022
(10% toward your final grade)

1. Sammy's Seashore Supplies rents beach equipment such as kayaks, canoes, beach chairs, and umbrellas to tourists.

a. Create a class to hold Rental data for Sammy's Seashore Supplies. The class contains:

- Two public final static fields that hold the number of minutes in an hour and the hourly rental rate (\$40)
- Four private nonstatic fields that hold a contract number, number of hours for the rental, number of minutes over an hour, and the price. The contract number is stored as a String because Sammy plans to assign contract numbers such as K681.
- Two public set methods. One sets the contract number (setContractNumber()). The other is named setHoursAndMinutes(), and it accepts the number of minutes for the rental and then sets the hours, extra minutes over an hour, and the total price. Note that the price is \$40 per hour plus \$1 for every extra minute.
- Four public get methods that return the values in the four nonstatic fields.
- Two overloaded constructors.
 - One constructor accepts a contract number and number of minutes as parameters. Pass these values to the setContractNumber() and setHoursAndMinutes() methods, respectively. The setHoursAndMinutes() method will automatically calculate the hours, extra minutes, and price.
 - The other constructor is a default constructor that passes "A000" and 0 to the two-parameter constructor. (use this() reference)

Save the above file as Rental.java.

(5 Marks)

b. Create a RentalDemo class to instantiate two Rental objects.

- Instantiate one object to retain the constructor default values.
- Accept user data for the contract number and minutes fields and use this data set to instantiate the second object.

(Hint: Create two static methods which prompt user to enter contract number and minutes, respectively (use Scanner class) and return these values to main() method in two local variables and then pass these variables to two-parameter constructor.

- Display all the details for both objects (use a static displayDetails() method)

Save the above file as RentalDemo.java.

(5 Marks)

2. Carly's Catering provides meals for parties and special events.
- a. Create an Event class for the company. The class contains:
- Two public final static fields that hold the price per guest (\$35) and the cutoff value for a large event (50 guests)
 - Three private fields that hold an event number, number of guests for the event, and the price. The event number is stored as a String because Carly plans to assign event numbers such as M312.
 - Two public set methods that set the event number (setEventNumber()) and the number of guests (setGuests()). The price does not have a set method because the setGuests() method will calculate the price as the number of guests multiplied by the price per guest every time the number of guests is set. *Figure out a way to use the cutoff value (50 guests) in an interesting manner.*
 - Three public get methods that return the values in three nonstatic fields.
 - Two overloaded constructors
 - One constructor accepts an event number and number of guests as parameters. Pass these values to the setEventNumber() and setGuests() methods, respectively. The setGuests() method will automatically calculate the event price.
 - The other constructor is a default constructor that passes "A000" and 0 to the two-parameter constructor. (use this() reference)

Save the above file as Event.java.

(5 Marks)

- b. Create an EventDemo class to instantiate two Event objects.
- Instantiate one object to retain the constructor default values.
 - Accept user data for the event number and guest fields, and use this data set to instantiate the second object. (Hint: Create two static methods which prompt user to enter event number and number of guests, respectively (use Scanner class) and return these values to main() method in two local variables and then pass these variables to two-parameter constructor.
 - Display all the details for both objects (use a static displayDetails() method)

Save the above file as EventDemo.java.

(5 Marks)

3. Create a class named Purchase. Each Purchase contains an invoice number, amount of sale, and amount of sales tax.
- Include set methods for the invoice number and sale amount. Within the set() method for the sale amount, calculate the sales tax as 5% of the sale amount. Also include a display method that displays a purchase's details. Save the file as **Purchase.java**.
 - Create an application that declares a Purchase object and prompts the user for purchase details. When you prompt for an invoice number, do not let the user proceed until a number between 1,000 and 8,000 has been entered. When you prompt for a sale amount, do not proceed until the user has entered a nonnegative value. (The validation code that enforces the restrictions on data fields should preferably reside inside the set methods and the other logic for testing can reside inside the main method). After a valid Purchase object has been created, display the object's invoice number, sale amount, and sales tax. Save the file as **CreatePurchase.java**.

(5 Marks)

Submission

- You need to submit your assignment through Blackboard community by the due date. NO LATE SUBMISSION will be allowed.
- You should do the assignment individually. *Any form of cheating or sharing of work may have serious consequences.*
- Your submission should be a .zip folder named a1-lastname_firstname.zip that contains: (1) A PDF file that has a cover page with Assignment title, Course name, Student name and Student ID, and Instructor name (2) All .java files
- Submit the assignment carefully. Only two attempts would be given, and the latest attempt would be graded.