CSSSKL142 – Lab10

Introduction to Classes

University of Washington Bothell

Summary

Today we'll experiment with software building blocks called classes. Recall that a class is a reusable unit of logic that encapsulates methods and data, and we're going to practice building small classes that can be assembled to compose more complex software containing other small classes. This is similar to building using Legos, these components are your building blocks that can produce complex models from combining individual elements.

Part 1

In this section we'll build a small class together, useful for managing a gift card balance. This class will keep track of the total balance and will allow for a starting total, deductions, and a balance report. To manage such details, we'll need a class-level (or instance) variable to store the balance, as well as a few methods to implement the actions mentioned above. Download **GiftCard.java** and look at the main method.

- Download the driver **GiftCard.java**.
- Add a class-level variable to track the balance at the top of the file. Should this variable be a char? A double?
- Add a method **void setBalance(double newBalance)** that resets the balance.
- Consider making this function fail on negative input
- Add a **deduct(double amount)** method that subtracts from the balance
- Consider making this function fail on a negative balance
- Add a **report()** method that prints out the balance remaining on the gift card.

Part 2

(*Warming-up to Objects*) Create a class named **Pizza** that stores information about **a single pizza**. It should contain the following:

- **Private instance variables** to store the size of the pizza (either small, medium, or large), the number of meat toppings, the number of veggie toppings and whether or not the pizza should be vegan-friendly (in which case you must make sure that the number of meat toppings is 0).
- **Constructor(s)** that set all of the instance variables. No-args defaults should be medium, no toppings, non-vegan.
- **Public methods** to get and set the instance variables.
- A public method named **calcCost** that returns a double that is the cost of the pizza. Pizza cost is determined by:
  - Small: $10
  - Medium: $12
  - Large: $14
  - Each meat topping cost $2 and each veggie topping cost $1
  - If the pizza is vegan, it costs an additional $2

- A public method named **getDescription** that returns a String containing the pizza size, whether it is vegan or not, quantity of each topping, and the pizza cost as calculated by **calcCost**.

In a class named **Pizzaria**, write test code to create several pizzas and output their descriptions. You can create the pizzas "manually" and test each pizza. For example, a large pizza with pepperoni and green pepper toppings should cost a total of $17. The description for such pizza would be as follows:

Large pizza, not vegan, 1 meat topping, 1 veggie topping: $17.00 dollars.

Part 3

(*A More Sophisticated Object*) Write a class called **Temperature** that has two instance variables: a **temp** value (a floating-point number - data type double) and a character for the **scale**, either 'C' for Celsius or 'F' for Fahrenheit. The class should have a constructor that sets each instance variable (assume zero degrees if no temp value is specified and Celsius if no scale is specified), and it should include the following methods:

1. **two accessor methods**: one to return the degrees Celsius and the other to return the degrees Fahrenheit— use the following formulas to write the two methods:

   ```
   DegreesC = 5(degreesF - 32)/9
   DegreesF = (9(degreesC)/5) + 32
   ```

2. **three mutator methods**: one to set the value, one to set the scale (F or C), and one to set both;

3. **a comparison method**: it should return -1 if the invoking temperature is less than the argument temperature, 0 if they are equal and 1 if the invoking temperature is greater than the argument one; and

4. **a toString method**: returns a String representing the calling Temperature object. The numeric value of the temperature should be rounded to the nearest tenth of a degree. For example,

   ```
   System.out.println(new Temperature(98.164555, F));    //should print: 98.2 F
   ```

Then write a driver program called **TempTester** that tests all the methods from the **Temperature** class. Example tests: 0.0 degrees C = 32.0 degrees F, -40.0 degrees C = -40.0 degrees F, and 100.0 degrees C = 212.0 degrees F.

Part 4

Don't forget to submit all of your .java files on Canvas by due date!

NO LATE SUBMISSIONS!