Project 1 - SCI Gas Station

CS 0401 — Intermediate Programming using Java

Check the Due Date on the CourseWeb

Open a new school is not easy. To ensure that School of Computer and Information (SCI) does not run into a financial problem, we decide to open a side business, a gas station. Since we are in SCI, who would be the best to write a program to control gas pumps other our own students from CS0401. So, for this project, you are going to help SCI by writing a program to control gas pumps. Luckily, we do not have that much money. So, our gas pumps are pretty low technology. In other words, they cannot accept a credit card. So, our gas station is a cash only business. Because of this, we do not have to deal with credit card companies. Besides, it is a lot easier to hide cash from government for tax purpose :). Oh, we also let our students who need extra money to wash cars during their free time. Our software will also incorporate students' car wash business as well.

For this project, you are going to write a program named `SCIGasStation.java`. Note that you may know about multi-file implementation in Java, but do not do it just yet. Just implement this project in one file. Having a multiple (helper) static methods may help organize your code for this project but it is not necessary for this project.

SCI Gas Station

For our SCI gas station, we have four type of gas based on octane ratings, 87, 89, 91, and 94. The price per gallon for each rating is shown in the table below:

Octane Rating	**Price per Gallon**
87	$2.59
89	$2.69
91	$2.84
94	$2.99

We also have four different types of car wash and the price of each type of car wash is shown below:

Type	**Price**
Basic	$5.00
Shine	$7.00
Deluxe	$9.00
Premium	$11.00

To encourage our customers to have their car washed, we give a 10-cent discount per gallon (up to 15 gallons) if they get both car wash and gas in one transaction. Note that we only give the discount for the first 15 gallons. After the first 15 gallons, it is the regular price. For example, if a customer get a car wash (Deluxe) as well as gas (87) and he/she purchases the total of 16.50 gallons. The total price will be as follows:

Car Wash (Deluxe)	$9.00
Gas (87) discount price	$2.49 * 15 = $37.35
Gas (87) regular price	$2.59 * 1.5 = $3.885
Total	$50.235 or $50.24

As we mentioned earlier, our gas pumps are old. So, they can only accept cash. Not only that, they can only accept four types of bank notes, Twenties, Tens, Fives, and Ones. But luckily, they can give changes back in bank notes as well as Quarters, Dimes, Nickels, and Pennies.

Part I: Gas Only or with Car Wash (10 Points)

First, your program should always greet customers and display all services that we offer as shown below:

```
Welcome to SCI Gas Station!!!
Please choose your service:
  1) Gas only
  2) Gas with car wash (10c off per gallon up to 15 gallons)
  3) Exit
```

Note that it is possible that a customer may enter a wrong option. If a customer enters a wrong option, your program should notify the customer and ask them to choose his/her choice again as shown below: Welcome to SCI Gas Station!!!

```
Please choose your service:
  1) Gas only
  2) Gas with car wash (10c off per gallon up to 15 gallons)
  3) Exit
4
Invalid option!!!
Please choose your service:
  1) Gas only
  2) Gas with car wash (10c off per gallon up to 15 gallons)
  3) Exit
```

At this part, if a customer choose option 3 (Exit), simple terminate your program.

Part II: Type of Gas and Car Wash (20 Points)

In this part, it depends on the option that a customer chose in Part I. As we mentioned earlier, the price of gas per gallon depends on the customer option. We also allow a customer to change their mind by going back to the main menu whenever he/she wants.

Gas Only

If a customer chose **Gas only**, simply display all type of gasoline that we offer together with their regular prices as shown below:

```
Please select a gasoline type:
  1) 87 ($2.59 per gallon)
  2) 89 ($2.69 per gallon)
  3) 91 ($2.84 per gallon)
  4) 94 ($2.99 per gallon)
  5) Cancel
```

As usual, make sure that the customer only enter a number between 1 and 5. Notify the customer if he/she enters an invalid option and display the menu again as shown below:

```
Please select a gasoline type:
  1) 87 ($2.59 per gallon)
  2) 89 ($2.69 per gallon)
  3) 91 ($2.84 per gallon)
  4) 94 ($2.99 per gallon)
  5) Cancel
6
Invalid option!!!
Please select a gasoline type:
  1) 87 ($2.59 per gallon)
  2) 89 ($2.69 per gallon)
  3) 91 ($2.84 per gallon)
  4) 94 ($2.99 per gallon)
  5) Cancel
```

Note that if a customer chooses option 5 (Cancel), simply go back to the main menu in Part I.

Gas with Car Wash

If a customer chooses gas as well as car wash, your program should as the customer to choose a type of car wash first followed by type of gasoline as shown below:

```
Please select a type of car wash:
  1) Basic ($5.00)
  2) Shine ($7.00)
  3) Deluxe ($9.00)
  4) Premium ($11.00)
  5) Cancel
3
Please select a gasoline type:
  1) 87 ($2.49 per gallon up to 15 gallons)
  2) 89 ($2.59 per gallon up to 15 gallons)
  3) 91 ($2.74 per gallon up to 15 gallons)
  4) 94 ($2.89 per gallon up to 15 gallons)
  5) Cancel
```

As usual, make sure that the customer only chooses the right option. Also note that the price per gallon for each gasoline rating is the regular price with the discount (up to 15 gallons).

Part III: Order Summery and Confirmation (10 Points)

After a customer successfully chose his/her service, your program should show the summary of their order as well as asking for a confirmation. The following are examples of a summary for gas only and a summary for gas and car wash:

```
You order is as follows:
  1. A full tank of gas 89 ($2.69 per gallon)
Confirm your order? (y/n):
```

```
You order is as follows:
  1. A full tank of gas 89 ($2.59 per gallon up to 15 gallons)
  2. Deluxe car wash ($9.0)
Confirm your order? (y/n):
```

Again, make sure that the customer enter either `y` or `n`. Notify the customer if he/she enter an invalid option and ask the customer to confirm his/her order again as shown below:

```
You order is as follows:
  1. A full tank of gas 89 ($2.69 per gallon)
Confirm your order? (y/n): Y
Invalid option!!!
Confirm your order? (y/n):
```

Note that if a customer enter `n`, simply go back to the main menu in Part I.

Part IV: Randomly Fill Up and Total (20 Points)

Once a customer confirms his/her order, your program is going to do the following:

1. Randomly generate a real number (`double`) between 10.0 and 20.0. This will be the amount of gallon to fill up the customer's car. Note that to generate a random number if Java is pretty simple. Consider the following code:

   ```
   import java.util.*;

   public class AClass
   {
       public static void main(String[] args)
       {
           Random rand = new Random();
           double aRandomNumber = rand.nextDouble();
       }
   }
   ```

   The variable `aRandomNumber` will be a random number (of type `double`) between 0.0 and 1.0. It is your job to modify it to be between 10.0 and 20.0. Note that if you need more random numbers, just simply call `rand.nextDouble()` again without doing `new Random()`.

2. Show the total value of the current transaction

For simplicity, tax is included in our price. Also be careful when you calculate the total. There are two prices for gasoline (regular and discount). The discount prices is for the first 15 gallons only. Here are some examples of output of this part, regular price, discount price with gasoline less than 15 gallons, and discount price with gasoline greater than 15 gallons:

```
You order is as follows:
  1. A full tank of gas 89 ($2.69 per gallon)
Confirm your order? (y/n): y
It took 12.52 gallons to fill up your car.
Your total is $33.68.
```

```
You order is as follows:
  1. A full tank of gas 89 ($2.59 per gallon up to 15 gallons)
  2. Deluxe car wash ($9.0)
Confirm your order? (y/n): y
It took 14.05 gallons to fill up your car.
Your total is $45.39.
```

```
You order is as follows:
  1. A full tank of gas 89 ($2.59 per gallon up to 15 gallons)
  2. Deluxe car wash ($9.0)
Confirm your order? (y/n): y
It took 17.53 gallons to fill up your car.
Your total is $54.67.
```

Note that examples above contain order summaries to help you verify your calculations.

Part V: Payments (20 Points)

In this part, your program is going to receive a cash payment from a customer. First, your program will show the customer how to enter a cash payment as shown below:

```
Please enter a payment amount in the following format:
  <amount><space><type>
    where <amount> is an integer
          <space> is a blank space
          <type> is either 20s, 10s, 5s, or 1s
  You may enter as many times as you like. Each entry will be
  added to your total until sufficient funds have been obtained.
Enter your payment:
```

At this point, your program is waiting for the customer to enter two entries separated by a single space that represents the number of bank notes and its type. For example, `2 20s` means two Twenties ($40), `3 5s` means three fives ($15), and so on. Note that the customer may enter as many time as he/she like until the total amount paid is greater than or equal the total from previous

part. Your program should also notify the customer how much money he/she just added to the total, the total amount paid, as well as the amount he/she owes until sufficient funds have been optioned. Here is an example:

```
Your total is $37.53.
Please enter a payment amount in the following format:
  <amount><space><type>
    where <amount> is an integer
          <space> is a blank space
          <type> is either 20s, 10s, 5s, or 1s
  You may enter as many times as you like. Each entry will be
  added to your total until sufficient funds have been obtained.
Enter your payment: 1 20s
    You have added $20.00 to your total.
    You have paid $20.00 out of $37.53.
    You still owe $17.53
Enter your payment: 1 10s
    You have added $10.00 to your total.
    You have paid $30.00 out of $37.53.
    You still owe $7.53
Enter your payment: 1 5s
    You have added $5.00 to your total.
    You have paid $35.00 out of $37.53.
    You still owe $2.53
Enter your payment: 3 1s
    You have added $3.00 to your total.
    You have paid $38.00 out of $37.53.
```

Note that `20s`, `10s`, and so on are not just a number. You should read them in as a `String`. Recall the `nextLine()` of the `Scanner` class. This will allow you to receive the keyboard input as a `String`. Do not forget that you need to use the `equals()` method of the class `String` to compare whether two `String` are identical. **For this part, we will assume that the user will always enter a valid input. In other words, the first number will be an integer greater than or equal to 1 and the second part will be a `String`, 20s, 10s, 5s, or 1s only**.

Part VI: Changes (10 Points)

As we mentioned earlier, your gas pumps can give back changes to customers. They have the ability to give changes in bank notes, Twenties, Tens, Fives, and Ones, as well as coins, Quarters, Dimes, Nickels, and Pennies. Note that our gas pumps should give changes back using the largest currency value possible. For example, for the change of $11.00, instead of two Fives and one One, our pumps should give one Ten and one One. Here are a couple of examples:

```
    You have paid $40.00 out of $38.03.
    Your change is $1.97.
      1 One
      3 Quarters
      2 Dimes
```

```
      2 Pennies
Thank you for shooping at SCI gas station
```

```
    You have paid $60.00 out of $43.48.
    Your change is $16.52.
      1 Ten
      1 Five
      1 One
      2 Quarters
      2 Pennies
Thank you for shooping at SCI gas station
```

Note that due to the precision of a floating-point number, there is a chance that your calculation may be off by one penny. Do not worry about that as long as it is only off by one penny. **Oh, since you already know about the `if` statement, you should be able to handle singular vs plural nouns**.

Part VII: Running Continuously (10 Points)

As in most gas station, once a transaction is completed, your program should go back to Part I.

Submission

The due date of this project is stated on the CourseWeb. Late submissions will not be accepted. You should submit the file `SCIGasStation.java` via CourseWeb.