**CIS611**
**Spring 2018**
**Individual Practice Programming Assignment: PA03**
Total Points: 20

***Iterations and Single Dimensional Arrays (Try_Catch for Input Validations):***

**The purpose of this programming assignment is to:**

- To control sequence of operations using loop
- To control loop using a sentinel value
- Understand the importance of arrays
- Initialize values in an array
- Program common array operations
- Understand the passing by reference concept
- Use appropriate naming conventions
- Follow programming style guidelines

Q1 (20 points):
You will write Java code to solve a problem on the rental for the ski resort and the total charge. You will write one Java class (name it as SkiRentalCalculation.java) to accomplish this. Within the main method of as SkiRentalCalculation.java, various code will be written to accomplish this. The input values need to be validated meaning, if no value is entered, or an empty space is entered, or an improper value is entered, the program will not crash.

Use 6 different JOptionPane to get the following input values

1. The program first asks for the total number of ski customers, in order to create **n** number of ski customers in an array. Use a JOptionPane to get this information.

For each ski customer, get the following pieces of information using different JOptionPane:

2. The name of the customer, which will hold both the first name and the last name of the ski customer. The acceptable value is John Doe. If no value is entered, or empty space is entered, or an invalid value is entered (e.g. abc), the program will not crash.

3. The number of the rental units for the ski customer. The acceptable values are 0, 1, 2, and 3. If no value is entered, or empty space is entered, or an invalid value is entered (e.g. abc), the program will not crash

4. Whether the ski customer is a first time user. The acceptable values are Yes, and No. If no value is entered, or empty space is entered, or an invalid value is entered (e.g. abc), the program will not crash

5. Whether the ski customer has a discount coupon. The acceptable values are Yes, and No. If

no value is entered, or empty space is entered, or an invalid value is entered (e.g. abc), the program will not crash

6. Whether the ski customer has a membership. The acceptable values are Yes, and No. If no value is entered, or empty space is entered, or an invalid value is entered (e.g. abc), the program will not crash.

Based on the above input values for each customer, use the table below to determine the total charge for each ski customer.

| Rate Category | # Rental items=0 | # Rental items=1 | # Rental items=2 | # Rental items=3 |
|---|---|---|---|---|
| Base rate per day. Multiply it by number of days | $60 | $110 | $155 | $190 |
| Membership discount | 5% | 8% | | |
| First Time User discount | 10% | 12% | | |
| Coupon discount | $5 | $10 | | |
| Sale tax | 7.5% | | | |

In the program, hard code the tax rate as 7.5%, as the same tax rate applies to all the customers, irrespective of the total charge. For this, in your program, code a final instance variable SALES_TAX_PERCENTAGE and initialized to 7.5%.

The code will be written so that the total charge of one customer at a time is calculated. The total charge computed may have a decimal value.

The customer name and the total charge needs to be stored in a new variable of data type String. You will define a data type String, which will hold the name of the customer and the total charge, which is computed. For the sake of simplicity, only these two values are stored in it, although it is possible to also hold the additional information of the number of rental items, whether the customer had a membership discount etc. This variable will have the name of the customer and the total charge concatenated. For example, the value stored I this String variable is “John Doe 175.73” You can format this String variable in any way you like, but this holds the basic information.

The code needs to be written so that the program has a loop to accept the information on the number of ski customers, received from the JOptionPane. The program finally displays the information on the ski customers in a new JOptionPane dialog box.

You will define an array of data type String, which will hold the information of each ski customer.

The fixed size of this skiCustomersArray is the same size, as the number entered in the JOptionPane for the number of ski customers. Each element of this array will hold information on the customer name and the total charge. For example, you will code similar to:

```
String numberOfSkiCustomers =JOptionPane.showInputDialog("Enter the number of ski
customers in whole number, e.g., 3");
int n = Integer.parseInt(numberOfSkiCustomers);
String[] skiCustomersArray = new String[n];
```

Where, n is the numeric value obtained from the JOptionPane on the number of the ski customers.


You will populate the value of each element of the skiCustomersArray array using the name of the ski customer and the total charge of each customer. For example, value of each element of this skiCustomersArray is similar to "John Doe, 175.73". You will concatenate the name of the ski customer, and the total charge to form the value of each element of this array. You can put a comma in between the name of the ski customer, and the total charge. Finally, display the information of all the ski customers (name of the customer and the total charge) in a new JOptionPane. You will loop through the elements of the skiCustomersArray to get the value of each element of this skiCustomersArray array. Put a new line separator between each customer in the displayed JOptionPane, so that the information for each customer is displayed in a new line. You can use something as a new line separator ("\n") after each customer information. The displayed information of the all the ski customers, with each customer information in a new line, inside the JOptionPane will be similar to:
John Doe, 175.73
Amy Doe, 181.56
Sandy Miller, 167.45

Any variation on the use of JOptionPane and the display is left to the creativity of the student.

***Evaluation Criteria:***

- The programs must compile cleanly (no compile errors, but compile warnings are sometimes accepted)
- The program should handles invalid data inputs by users and terminates gracefully
- The programs should not crash while running and it should terminate
- All tasks (requirements) in this assignment must be completed in order to receive credit
- The correct understanding and implementation (coding) of the requirements (programs should behave as anticipated):
    - The programs must terminate with proper/correct outputs

- All the logical computations should be performed correctly

***Submission: (This is an individual Assignment!)***

Copy the .java source files from the *src* folder in your *work space* to another folder that should be named following the provided naming format in this course, then zip and upload the file under this assignment answer in Canvas.

***File Name:*** *FLLLLPA03.zip (F = first letter in your first name and LLLL = your last name)*

## Grading Rubric PA03

**Student Name: _____**

**Question 1**

| Requirements | Comment | Max Points Allowed | Points Earned |
|---|---|---|---|
| General Code Structure:<br><br>Creating a Java file with name SkiRentalCalculation.java (0.25)<br><br>Comments used in the code to explain the purpose of the code (0.25)<br><br>Indentation of the code for better readability (0.25)<br><br>Good choice of variable names (0.25) | | 1 | |
| Input, Output, User Interface:<br><br>Proper coding implementation of JOptionPane to accept the 6 input values using 6 different JOptionPane (4)<br><br>Exception handling of the invalid input values using try-catch. If no value is entered, or empty space is entered, or character value is entered (when numeric is expected), the program should not crash (3)<br><br>Proper display of the customer name and the total charge in a JOptionPane. The total charge may have decimal value. Put a line separator between each customer information (2) | | 9 | |
| General Algorithm and Logic:<br><br>Proper calculation of total charge based on the various input values, and the Rate category table included in this assignment (3.5) | | 10 | |

| | | | |
|---|---|---|---|
| Use of array to hold the customer name and the total charge of the customers (3.5)<br><br>Logic to iterate through the array and display the information of the customers (customer name and the total charge) in the JOptionPane (3) | | | |
| Total | | 20 | |

**Total ____/20**