CSCI 1100 – Fall 2016
Assignment 3 – Due Friday December 2 at 11:00 pm (evening)
Submit on Brightspace

Name: Jeremy Peters

Student ID: B00707976

Declaration: Please complete this declaration		
1	"This document is entirely my own work." If no, acknowledge any assistance below; outside help should only be used to help you understand the questions NOT to provide the solutions.	Yes/no.
2	I obtained help to complete this document (e.g., from a TA).	Yes/no. If Yes, give Details. T.A. Xinlun Liu helped me answer a question.
3	This document contains some guidance from the Internet or another document or file or program (e.g., Java's API).	Yes/no. If Yes, give details and provide references.

Exercise 1. Write a program that asks a user to enter in two words. The program then calls a method called *uniqueCharacters* that takes two Strings as parameters and prints the characters that are unique to both the Strings (i.e., **not in both Strings**). [You can use the String method lastIndexOf (char) to help you.] The order of the characters that you print does not matter. See Sample output below. You should have at least 3 test cases (different from the sample).

Sample:

```
Enter a word: abccd
Enter a word: ceccaa
Unique character/s: b e d
```

Program:

```
/*CSCI 1100-Assignment 3 -"Question 1"
This program will find characters in two user-entered strings that
are unique to each string.
<Jeremy Peters><B00707976> <Nov 6, 2016>*/
```

```java
import java.util.Scanner;
public class A3Q1{
   public static void main(String[] args){
      Scanner keyboard = new Scanner(System.in);
      String input1, input2, result = " ";
      //will store user inputs and final result
      System.out.print("Enter a word: ");//asks for first input
      input1 = keyboard.nextLine();//stores first input
      System.out.print("Enter a word: ");//asks for second input
      input2 = keyboard.nextLine();//stores second input
      result = uniqueCharacters(input1, input2, result);
      /*This method will take each character from the first string
      and determine if that character is unique to the first string.
      The result will store all such unique characters.*/
      result = uniqueCharacters(input2, input1, result);
      /*Since we want unique characters for both user inputs,
      reversing the arguments for the method call will ensure the
      second string will have its characters searched.*/
      if (result.equals(" "))
      //tells user if no unique characters are found
         System.out.print("No unique characters");
      else
      System.out.print("Unique character/s:" + result);
      //prints final result if there are unique characters.
   }
   public static String uniqueCharacters(String word1, String word2,
   String unique){/*This method will update the string "unique" with
   characters in the first string not found in the second string.*/
      char ch1;//selects a character in word1
      int index2, indexUnique;
      /*This will determine whether a character from word1 is found
      in word2 or the string "unique."*/
      for(int i=0; i<word1.length(); i++){
      //i determines which character in 1st word is analyzed.
         ch1 = word1.charAt(i);//stores a character in 1st word
         index2 = word2.indexOf(ch1, 0);//finds 'ch1' in 2nd word
         indexUnique = unique.indexOf(ch1, 0);
         //finds 'ch1' in the string "unique"
         if(index2==-1 && indexUnique==-1)
         /*if 'ch1' is not present in the second word, and it is not
         already recorded in the string "unique," ch1 is added to
         the string "unique."*/
            unique+=ch1+" ";//updates "unique"
      }
      return unique;
      /*returns the string which contains the unique characters to
      the first string.*/
   }
}
```

Output:

```
  ----jGRASP exec: java A3Q1
Enter a word: afbbccddeeaf
Enter a word: ghbbccddeegh
Unique character/s: a f g h
  ----jGRASP: operation complete.

  ----jGRASP exec: java A3Q1
Enter a word: Jeremy
Enter a word: Peters
Unique character/s: J m y P t s
  ----jGRASP: operation complete.

  ----jGRASP exec: java A3Q1
Enter a word: word
Enter a word: word
No unique characters
  ----jGRASP: operation complete.
```

Exercise 2. You can represent a route map in a Java program with two arrays. The first array can contain the names of the towns on the route and the second can contain the distances between the towns. For example, you could represent the route from Halifax to Toronto with the following two arrays:

String[ ] city = {"Halifax", "Moncton", "Fredericton", "Quebec City", "Montreal", "Toronto"}
int[ ] distance = { 260, 180, 520, 250, 504, 0};

As well a third array can hold the price of gas per litre in each city.

double [ ] price = {1.1, 1.0, 1.15, 0.98, 1.01, 0}

Assume that your car holds 40 L of gas and the car drives 13 kilometers per litre (KMPL).

Based on these arrays, to go from Halifax to Moncton it is 260 km. To go 260 km, you need 20 L of gas (distance/KMPL). The price of gas in Halifax is $1.10/L and the price of gas in Moncton is $1.00/L.

Write a program that will work out how much gas (in Litres) you should buy at each city to pay the least when traveling from Halifax to Toronto (assume that you only know the price of gas at the current city and the next city you are travelling to). Note, you do not need to use methods for this question (that is, you can write the program all in the main method).

For the above data the program would print (you need to deal with decimals places – maximum 2 decimal places):

Buy gas as follows:

Halifax: Buy 20 litres Pay $22.00
Moncton: Buy 40 litres Pay $40.00
Fredericton: Buy XX litres Pay $xx.xx
Quebec City: Buy XX litres Pay $xx.xx
Montreal: Buy XX litres Pay $xx.xx
Toronto = Buy XX litres Pay $xx.xx
You purchased X Litres and the Total Cost would be $ xx.xx

Test your program with this data and with two other sets of data different than above.

Test1 Program (Same data as in sample):

```
/*CSCI 1100-Assignment 3 -"Question 2"
This program will calculate the amount of gas required to travel
from Halifax to Toronto, and the resulting cost.
<Jeremy Peters><B00707976> <Nov 6, 2016>*/

public class A3Q2{
   public static void main(String[] args){
      String[ ] city = {"Halifax", "Moncton", "Fredericton",
      "Quebec City", "Montreal",  "Toronto"};
      //Stores the stopping points along the trip.
      int[] distance = {260, 180, 520, 250, 504, 0};
      //Stores distances from current location to the next stop.
      double [] price = {1.1, 1.0, 1.15, 0.98, 1.01, 0};
      //Stores the price of gas per litre at each location.
      double buyGas, cost, amountOfGas, kmpl, maxGas;
      maxGas = 40;//Stores maximum fuel capacity (litres)
      kmpl = 13;//Stores fuel efficiency (kilometres per litre)
      amountOfGas = 0;//Stores amount of gas currently in tank.
      double totalGas = 0, totalCost = 0;
      /*TotalGas measures total gas consumed.  totalCost measures
      the total cost of buying the gas.*/
      System.out.print("Buy gas as follows:\n");
      for(int i=0; i<city.length-1; i++){
      //Computes gas needed at each stopping point along the trip.
         if(price[i]<price[i+1])
            buyGas = maxGas - amountOfGas;
            /*buyGas stores the recommended amount of gas the user
            should buy at that given stop.*/
         /*This "if" statement  tells the program to capitalize on
         the cheaper price of gas at this current location.*/
         else
            buyGas = (distance[i]/kmpl) - amountOfGas;
         /*This "else" statement tells the program to only buy the
         necessary amount of gas to get to the next stop.*/
         amountOfGas+=buyGas;//tank is refilled and updated.
         cost = buyGas*price[i];
         /*stores the cost of the gas the user will buy*/
         totalGas+=buyGas;//updates total gas
         totalCost+=cost;//updates total cost
```

```
            System.out.printf(city[i]+": Buy %.0f litres Pay $%.2f\n",
            buyGas, cost);//prints gas purchased and cost at each stop.
            amountOfGas-=(distance[i]/kmpl);
            /*Gas is used up after leaving current location.*/
         }
         System.out.printf("You purchased %.0f Litres and the Total "+
                           "Cost would be $%.2f", totalGas, totalCost);
         //prints total amount of gas bought at each stop.
      }
}
```

Test1 Output:

```
 ----jGRASP exec: java A3Q2
Buy gas as follows:
Halifax: Buy 20 litres Pay $22.00
Moncton: Buy 40 litres Pay $40.00
Fredericton: Buy 14 litres Pay $15.92
Quebec City: Buy 40 litres Pay $39.20
Montreal: Buy 18 litres Pay $18.18
You purchased 132 Litres and the Total Cost would be $135.30
 ----jGRASP: operation complete.
```

Test2 Program:

```
/*CSCI 1100-Assignment 3 -"Question 2"
This program will calculate the amount of gas required to travel
from Halifax to Toronto, and the resulting cost.
<Jeremy Peters><B00707976> <Nov 6, 2016>*/

public class A3Q2{
   public static void main(String[] args){
      String[ ] city = {"Halifax", "Moncton", "Fredericton",
      "Quebec City", "Montreal",  "Toronto"};
      //Stores the stopping points along the trip.
      int[] distance = {300, 150, 450, 280, 520, 0};
      //Stores distances from current location to the next stop.
      double [] price = {1.2, 0.95, 0.90, 1.0, 1.15, 0};
      //Stores the price of gas per litre at each location.
      double buyGas, cost, amountOfGas, kmpl, maxGas;
      maxGas = 40;//Stores maximum fuel capacity (litres)
      kmpl = 13;//Stores fuel efficiency (kilometres per litre)
      amountOfGas = 0;//Stores amount of gas currently in tank.
      double totalGas = 0, totalCost = 0;
      /*TotalGas measures total gas consumed.  totalCost measures
      the total cost of buying the gas.*/
      System.out.print("Buy gas as follows:\n");
      for(int i=0; i<city.length-1; i++){
      //Computes gas needed at each stopping point along the trip.
         if(price[i]<price[i+1])
            buyGas = maxGas - amountOfGas;
```

```
            /*buyGas stores the recommended amount of gas the user
            should buy at that given stop.*/
         /*This "if" statement  tells the program to capitalize on
         the cheaper price at this current location.*/
         else
            buyGas = (distance[i]/kmpl) - amountOfGas;
         /*This "else" statement tells the program to only buy
         necessary amount of gas to get to the next stop.*/
         amountOfGas+=buyGas;//tank is refilled and updated.
         cost = buyGas*price[i];
         /*stores the cost of the gas the user will buy*/
         totalGas+=buyGas;//updates total gas
         totalCost+=cost;//updates total cost
         System.out.printf(city[i]+": Buy %.0f litres Pay $%.2f\n",
         buyGas, cost);//prints gas purchased and cost at each stop.
         amountOfGas-=(distance[i]/kmpl);
         /*Gas is used up after leaving current location.*/
      }
      System.out.printf("You purchased %.0f Litres and the Total "+
                        "Cost would be $%.2f", totalGas, totalCost);
      //prints total amount of gas bought at each stop.
   }
}
```

Test2 Output:

```
 ----jGRASP exec: java A3Q2
Buy gas as follows:
Halifax: Buy 23 litres Pay $27.69
Moncton: Buy 12 litres Pay $10.96
Fredericton: Buy 40 litres Pay $36.00
Quebec City: Buy 35 litres Pay $34.62
Montreal: Buy 22 litres Pay $24.77
You purchased 131 Litres and the Total Cost would be $134.04
 ----jGRASP: operation complete.
```

Test3 Program:

```
*CSCI 1100-Assignment 3 -"Question 2"
This program will calculate the amount of gas required to travel fro
from Halifax to Toronto, and the resulting cost.
<Jeremy Peters><B00707976> <Nov 6, 2016>*/

public class A3Q2{
   public static void main(String[] args){
      String[ ] city = {"Halifax", "Moncton", "Fredericton",
      "Quebec City", "Montreal",  "Toronto"};
      //Stores the stopping points along the trip.
      int[] distance = {500, 200, 400, 195, 490, 0};
      //Stores distances from current location to the next stop.
      double [] price = {0.8, 1.0, 1.2, 1.1, 1.1, 0};
```

```
        //Stores the price of gas per litre at each location.
        double buyGas, cost, amountOfGas, kmpl, maxGas;
        maxGas = 40;//Stores maximum fuel capacity (litres)
        kmpl = 13;//Stores fuel efficiency (kilometres per litre)
        amountOfGas = 0;//Stores amount of gas currently in tank.
        double totalGas = 0, totalCost = 0;
        /*TotalGas measures total gas consumed.  totalCost measures
        the total cost of buying the gas.*/
        System.out.print("Buy gas as follows:\n");
        for(int i=0; i<city.length-1; i++){
        //Computes gas needed at each stopping point along the trip.
           if(price[i]<price[i+1])
              buyGas = maxGas - amountOfGas;
              /*buyGas stores the recommended amount of gas the user
              should buy at that given stop.*/
           /*This "if" statement  tells the program to capitalize on
           the cheaper price at this current location.*/
           else
              buyGas = (distance[i]/kmpl) - amountOfGas;
           /*This "else" statement tells the program to only buy
           necessary amount of gas to get to the next stop.*/
           amountOfGas+=buyGas;//tank is refilled and updated.
           cost = buyGas*price[i];
           /*stores the cost of the gas the user will buy*/
           totalGas+=buyGas;//updates total gas
           totalCost+=cost;//updates total cost
           System.out.printf(city[i]+": Buy %.0f litres Pay $%.2f\n",
           buyGas, cost);//prints gas purchased and cost at each stop.
           amountOfGas-=(distance[i]/kmpl);
           /*Gas is used up after leaving current location.*/
        }
        System.out.printf("You purchased %.0f Litres and the Total "+
                          "Cost would be $%.2f", totalGas, totalCost);
        //prints total amount of gas bought at each stop.
    }
}
```

Test3 Output:

```
 ----jGRASP exec: java A3Q2
Buy gas as follows:
Halifax: Buy 40 litres Pay $32.00
Moncton: Buy 38 litres Pay $38.46
Fredericton: Buy 6 litres Pay $7.38
Quebec City: Buy 15 litres Pay $16.50
Montreal: Buy 38 litres Pay $41.46
You purchased 137 Litres and the Total Cost would be $135.81
 ----jGRASP: operation complete.
```