**CSCI 1100 – Fall 2016**

**Assignment 2 – Due Tuesday Nov. 15 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.  My father, Gregory Peters, helped me fix one logic error based on the differences in using keyboard.next() and keyboard.nextLine() in my programming for question 4. I understand that the keyboard.next() reads until the next space in a user input, and keyboard.nextLine() reads until the enter key. This difference was what caused my logic error. |
| 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.  I used Java’s API to learn about the indexOf method.  https://docs.oracle.com/javase/7/docs/api/ |

**Exercise 1.** Write a program that asks a user to enter a number. Your program will then print out all the *positive factors* of that number (i.e., numbers that divide evenly into the number, leaving no remainder). See the sample code below. Use a while loop for this question. You should have 3 test cases (all different from the sample).

**Sample:**

Enter a number: 6

Factors: 1 2 3 6

Enter a number: 12

Factors: 1 2 3 4 6 12

Enter a number: 3

Factors: 1 3

**Program:**

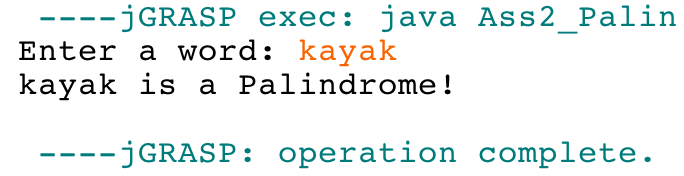
/\*CSCI 1100-Assignment 2-"Question 1"  
This program will list all positive factors of a user-entered  
number.  
<Jeremy Peters><B00707976> <Nov 6, 2016>\*/  
  
import java.util.Scanner;  
public class A2Q1{  
 public static void main(String[] args){  
 Scanner keyboard = new Scanner(System.in);  
 System.out.print("Enter a number: ");  
 int n = keyboard.nextInt(), i=1;  
 System.out.print("Factors: ");  
 //n is the entered number, and i is a counter.  
 while(i<=n){//Counter ranges from 1 to n.  
 if(n%i==0)//n is divisible by the counter.   
 System.out.print(i+" ");  
 /\*If n is divisible by the counter, the counter is a  
 factor, and will be outputted.\*/   
 i++;//Updates counter.  
 }  
 }  
}

**Output:**  
 ----jGRASP exec: java A2Q1  
Enter a number: 24  
Factors: 1 2 3 4 6 8 12 24   
 ----jGRASP: operation complete.  
  
 ----jGRASP exec: java A2Q1  
Enter a number: 25  
Factors: 1 5 25   
 ----jGRASP: operation complete.  
  
 ----jGRASP exec: java A2Q1  
Enter a number: 23  
Factors: 1 23   
 ----jGRASP: operation complete.

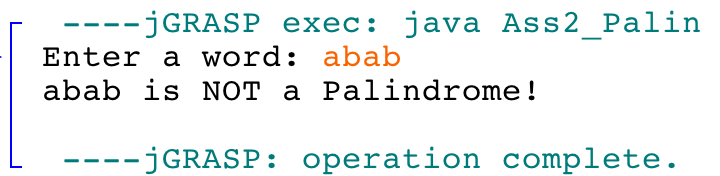
**Exercise 2.** Write a program that asks a user to enter a word. The program then tests to see if the word is the same spelled forward as backward (i.e., whether or not the word is a 'palindrome'). It prints a message indicating whether the word is a palindrome or not. Use a for loop. See the sample output below. You should have 3 test cases (all different from the samples provided here).

**Sample output:**

**//Shows when a word is a palindrome**

****

**//shows when a word is not a palindrome**

****

**Program:**

/\*CSCI 1100-Assignment 2-"Question 2"  
This program will determine whether a user-entered word is a   
palindrome (reads the same forwards and backwards).  
<Jeremy Peters><B00707976> <Nov 6, 2016>\*/  
  
import java.util.Scanner;  
public class A2Q2{  
 public static void main(String[] args){  
 Scanner keyboard = new Scanner(System.in);  
 System.out.print("Enter a word: ");  
 String word = keyboard.next();//Stores entered word.  
 String lwrWord = word.toLowerCase();  
 int n=word.length();   
 /\*Changes word to lowercase. The definition of a   
 palindrome ignores cases.\*/  
 System.out.print(word+" is");  
 /\*The final output is split up so that the program can  
 inject the word "not" into the phrase "is a palindrome,"  
 if necessary.\*/  
 for(int i=0; i<=n/2; i++){  
 if(lwrWord.charAt(i)!=lwrWord.charAt(n-1-i)){  
 System.out.print(" NOT");  
 i=(n/2)+1;//Exits the loop by making the test false.  
 }   
 }/\*This for loop will match the first character with   
 the last character, the second character with the second   
 last character, and so on, until the middle of the word.  
 If, at any point, the corresponding characters don't   
 match, the word is not a palindrome, the program will   
 change the output accordingly, and prematurely exit the   
 loop.\*/   
 System.out.print(" a Palindrome!");  
 }  
}

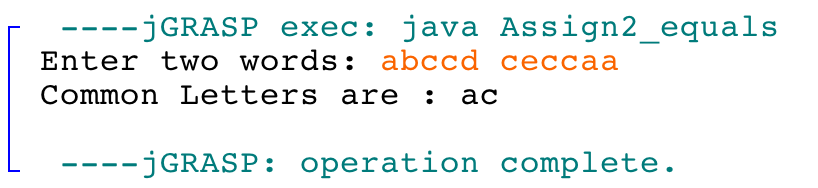
**Output:**

 ----jGRASP exec: java A2Q2  
Enter a word: racecar  
racecar is a Palindrome!  
 ----jGRASP: operation complete.  
  
 ----jGRASP exec: java A2Q2  
Enter a word: ABBA  
ABBA is a Palindrome!  
 ----jGRASP: operation complete.  
  
 ----jGRASP exec: java A2Q2  
Enter a word: powwow  
powwow is NOT a Palindrome!  
 ----jGRASP: operation complete.

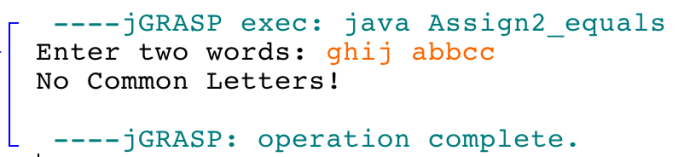
**Exercise 3.** Write a program that asks a user to enter in two words. The program prints a string consisting of the characters that are common to both the words (letters that are in both words). [You should look up the String method indexOf (char) in the Java API documentation to help you.] Repeated letters must be counted only once, and the order of common letters is not important. For example, if the Strings are abccd and ceccaa the value of the String returned by the method would be ac (order of characters not significant). Use a for loop. See Sample output below. You should have at least 3 test cases (different from the samples).

**Sample output:**

**//Shows when two words have common letters**

****

**//shows when two words have no letters in common**

****

**Program:**

/\*CSCI 1100-Assignment 2-"Question 3"  
This program will find the matching characters between two user-  
entered words.   
<Jeremy Peters><B00707976> <Nov 6, 2016>\*/  
  
import java.util.Scanner;  
public class A2Q3{  
 public static void main(String[] args){  
 Scanner keyboard = new Scanner(System.in);  
 System.out.print("Enter two words: ");  
 String word2 = keyboard.next(), word1 = keyboard.next();  
 //Two inputted words are stored.  
 int n1 = word1.length(), i, j=0;  
 /\*n1 is the length of word1, i and j are counters   
 (see future comments).\*/  
 for(i=n1-1;i>=0;i--){  
 /\*i corresponds to the character positions of the first  
 entered word (word1). Note that i starts at the last   
 character, and ends at the first character of word1.\*/  
 char ch1 = word1.charAt(i);  
 /\*The character in word1 at position i will be  
 referred to the "character of interest."\*/  
 int index2 = word2.indexOf(ch1,0);  
 /\*This determines where (or if) the character of   
 interest occurs in word2, the second entered word.\*/  
 int index1 = word1.indexOf(ch1,i+1);  
 /\*This will determine if the character of interest   
 occurs after position i in word1. If this is the case,  
 then the program has encountered the character of  
 interest, because the program is working backwards.\*/   
 if(index2!=-1 && index1==-1){  
 /\*The character of interest is found in word2, and  
 has not already been encountered in word1.\*/  
 if(j==n1-1-i)  
 System.out.print("Common Letters are : ");  
 /\*j counts the number of characters in word1 that  
 were encountered and not matched to any characters  
 in word2. When the program starts, i and j add up  
 to n-1, until (and including) the finding of the   
 first match. Since a match was found, and i+j=n1-1,  
 this match is the first match, and the output is   
 justified.\*/   
 System.out.print(ch1);//prints character of interst.  
 }   
 else//Character of interest is not in word2  
 j+=1;//Counter of non-matching letters updates by 1.  
 }  
 if(j==n1)  
 System.out.print("No Common Letters!");  
 /\*Since j counts the number of non matching letters,   
 no matching characters were found if j is the length   
 of word1.\*/  
 System.out.println();   
 }  
}

**Output:**

 ----jGRASP exec: java A2Q3  
Enter two words: abcdef gbijkl  
Common Letters are : b  
  
 ----jGRASP: operation complete.  
  
 ----jGRASP exec: java A2Q3  
Enter two words: Jeremy Peters  
Common Letters are : re  
  
 ----jGRASP: operation complete.  
  
 ----jGRASP exec: java A2Q3  
Enter two words: the quick  
No Common Letters!  
  
 ----jGRASP: operation complete.

**Exercise 4.** Write a program that calculates the shipping fee for an online shoe store. The program will prompt the user to enter the number of pairs of shoes they wish to order. Then the program will ask the user to enter the cost of each pair of shoes. The program will calculate the total cost of the shoes and apply a shipping fee based on this total cost. The program will ask the customer to enter where they want to ship the order: Nova Scotia, Canada (excluding Nova Scotia), or Other (US/International). If the program does not recognize the shipping destination, the program will keep asking until the user enters a correct location. Then the program will determine an additional shipping fee based on where the customer wants to ship the order. For example, if the shoes are to be shipped within Nova Scotia there is no additional fee, but to send the order elsewhere in Canada would cost an additional $25. You can use for loops **or** while loops **or** a combination of both for this exercise. You **need** to generate correct a monetary output format (i.e. 2 decimal places) in this question. See the end of the assignment for a method (printf) that will help you control the places after the decimal.

See below for shipping costs:

Shipping fee based on cost:

An order that costs less than $100: 25% of the total

An order that costs between $100 dollars and $200: 15% of the total

Orders that cost more than $200: 10% of the total

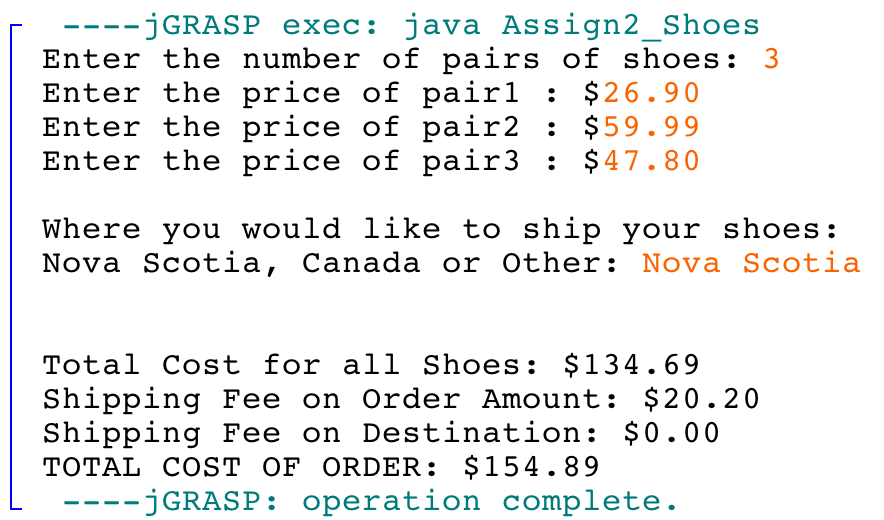
Added shipping fee based on destination:

Shipping within Nova Scotia: no additional charge

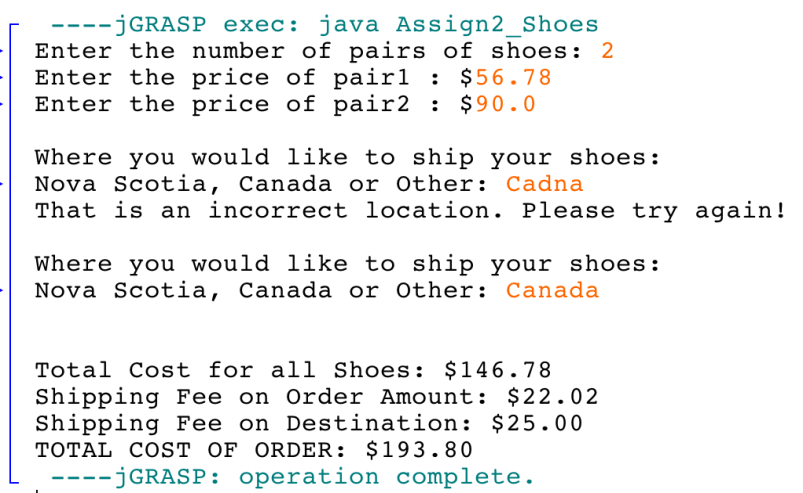
Shipping elsewhere in Canada: $25.00

Shipping outside Canada: $50.00

**Some sample runs:**

**//Shows costs for three pairs of shoes being shipped to NS**

**//Shows cost of two pairs of shoes being shipped in Canada with a corrected location**



**Program:**

/\*CSCI 1100-Assignment 2-"Question 4"  
This program will calculate the shipping cost for a shoe order.  
<Jeremy Peters><B00707976> <Nov 6, 2016>\*/  
  
import java.util.Scanner;  
public class A2Q4{  
 public static void main(String[] args){  
 Scanner keyboard = new Scanner(System.in);  
 System.out.print("Enter the number of pairs of shoes: ");  
 int n = keyboard.nextInt();//records number of shoes.  
 double sumOfPrices, costShippingFee, p, locShippingFee;  
 String shippingLocation;  
 /\*sumOfPrices is the subtotal of each user-entered price.  
 costShippingFee is the shipping fee based on the subtotal.  
 p is the percentage of the total cost that determines the  
 shipping fee based on cost.  
 locShippingFee is the shipping fee based on the destination.  
 Shipping location is the destination itself.\*/   
 sumOfPrices = 0;  
 for(int i=1;i<=n;i++){  
 /\*Reads the price of each shoe pair, and updates the total\*/  
 System.out.print("Enter the price of pair"+i+" : $");  
 sumOfPrices+=keyboard.nextDouble();  
 }  
 if(sumOfPrices<100)//subtotal is less than $100  
 p = 0.25;//cost-based shipping fee is 25% of the subtotal.  
 else if(sumOfPrices>=100 && sumOfPrices<=200)  
 //Subtotal is between $100 and $200  
 p = 0.15;//cost-based shipping fee is 15% of the subtotal.  
 else//Subtotal is greater than $200  
 p = 0.10;//cost-based shipping fee is 10% of the subtotal.  
 costShippingFee = p\*sumOfPrices;  
 System.out.print("\nWhere would you like to ship your sho");  
 System.out.print("es:\nNova Scotia, Canada, or Other: ");  
 keyboard.nextLine();  
 shippingLocation = keyboard.nextLine();  
 while((shippingLocation.equals("Nova Scotia")==false) &&  
 (shippingLocation.equals("Canada")==false) &&  
 (shippingLocation.equals("Other")==false)){  
 System.out.print("That is an incorrect location. " +   
 "Please try again! ");   
 shippingLocation = keyboard.nextLine();  
 /\*This while loop ensures the user inputs only "Nova Scotia,"  
 "Canada," or "Other," which are used in if statements.\*/   
 }   
 if(shippingLocation.equals("Nova Scotia"))  
 locShippingFee = 0;  
 /\*No additional shipping fee if Nova Scotia is destination\*/  
 else if(shippingLocation.equals("Canada"))  
 locShippingFee = 25;  
 /\*$25 additional shipping fee if the shipping destination is   
 anywhere in Canada except Nova Scotia.\*/   
 else  
 locShippingFee = 50;  
 /\*$50 additional shipping fee for outside Canada.\*/   
 /\*The following outputs are controlled to 2 decimal places.\*/  
 System.out.printf("Total cost for all shoes: $%.2f",  
 sumOfPrices);//prints subtotal  
 System.out.printf("\nShipping fee on order amount: $%.2f",  
 costShippingFee);//cost-based shipping.   
 System.out.printf("\nShipping fee on destination: $%.2f",  
 locShippingFee);//destination shipping.  
 double grandTotal = sumOfPrices + costShippingFee +  
 locShippingFee;  
 System.out.printf("\nTOTAL COST OF ORDER: $%.2f",  
 grandTotal);//prints final total.   
 }  
}

**Output:**  
 ----jGRASP exec: java A2Q4  
Enter the number of pairs of shoes: 1  
Enter the price of pair1 : $80.00  
  
Where would you like to ship your shoes:  
Nova Scotia, Canada, or Other: USA  
That is an incorrect location. Please try again!   
  
Where would you like to ship your shoes:  
Nova Scotia, Canada, or Other: Arizona  
That is an incorrect location. Please try again!   
  
Where would you like to ship your shoes:  
Nova Scotia, Canada, or Other: Other  
  
Total cost for all shoes: $80.00  
Shipping fee on order amount: $20.00  
Shipping fee on destination: $50.00  
TOTAL COST OF ORDER: $150.00  
 ----jGRASP: operation complete.  
  
 ----jGRASP exec: java A2Q4  
Enter the number of pairs of shoes: 2  
Enter the price of pair1 : $100  
Enter the price of pair2 : $49.9999  
  
Where would you like to ship your shoes:  
Nova Scotia, Canada, or Other: CAN  
That is an incorrect location. Please try again!   
  
Where would you like to ship your shoes:  
Nova Scotia, Canada, or Other: Canada  
  
Total cost for all shoes: $150.00  
Shipping fee on order amount: $22.50  
Shipping fee on destination: $25.00  
TOTAL COST OF ORDER: $197.50  
 ----jGRASP: operation complete.  
  
 ----jGRASP exec: java A2Q4  
Enter the number of pairs of shoes: 4  
Enter the price of pair1 : $50.25  
Enter the price of pair2 : $100.25  
Enter the price of pair3 : $49.249  
Enter the price of pair4 : $100  
  
Where would you like to ship your shoes:  
Nova Scotia, Canada, or Other: Nova Scotia  
  
Total cost for all shoes: $299.75  
Shipping fee on order amount: $29.97  
Shipping fee on destination: $0.00  
TOTAL COST OF ORDER: $329.72  
 ----jGRASP: operation complete.