

CSCI 1100 – Fall 2016
Assignment 2 – Due Tuesday Nov. 15 at 11:00 pm (evening)
Submit on Brightspace

Name: Mihyar Al-Massalma

Student ID: B00759975

Assignments are to be your own work. If you have questions, you can ask your Instructor, course TAs, or TAs in the Learning Centre.

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
2	I obtained help to complete this document (e.g., from a TA).	No
3	This document contains some guidance from the Internet or another document or file or program (e.g., Java's API).	No

*Your task is to complete this report using Word (or another word processing program) and JGrasp and to submit the complete Word document on Brightspace **saved as a pdf**. Read the questions carefully! Acknowledge any help that you obtained from your Lab or Learning Centre TAs in the table above. **Again note, assignments must be your own work.***

- **Submit using Brightspace. Make sure you double check that the file has been added to Brightspace.**
- TAs can only provide help in understanding the problem and by giving other small hints.
- Make sure your programming code is neatly formatted and properly commented (you will lose marks for poor formatting and commenting).
- Also, be sure to provide the requested number of test outputs. Each test output must be different from the sample outputs given for each question.

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

```
/* Assignment2 Question 1
This program will give back the factors of a number
<CSCI 1100><Mihyar Al-Masalma><B00759975>*/
import java.util.Scanner; // import Scanner class
public class Q1 {
    public static void main(String[] args) {
        int counter = 1;
        String result = "";
        // Create an instance of the Scanner class
        Scanner input = new Scanner(System.in);
        // Ask the user to key-in a number
        System.out.print("Enter a number: ");
        // Store the number in a variable
        int number = input.nextInt();
        // Check the number starting from 1 to the number
        while (number >= counter) {
            // Check if the number is dividable
            if (number % counter == 0) {
                // Add the number to the result
                result += counter;
                // add a space between numbers
                result += " ";
            }
            // Increase the counter
            counter++;
        }
        // Print out the result
        System.out.print("Factors: "+result);
    }
}
```

```

> ----jGRASP exec: java Q1
> Enter a number: 144
> Factors: 1 2 3 4 6 8 9 12 16 18 24 36 48 72 144
> ----jGRASP: operation complete.
```

```

> ----jGRASP exec: java Q1
> Enter a number: 256
> Factors: 1 2 4 8 16 32 64 128 256
> ----jGRASP: operation complete.
```

```

    ----jGRASP exec: java Q1
▶ Enter a number: 14
  Factors: 1 2 7 14
    ----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
----jGRASP exec: java Ass2_Palin
Enter a word: kayak
kayak is a Palindrome!

----jGRASP: operation complete.

```

```

//shows when a word is not a palindrome
    ----jGRASP exec: java Ass2_Palin
    Enter a word: abab
    abab is NOT a Palindrome!

    ----jGRASP: operation complete.

```

```

/* Assignment2 Question 2
This program will reverse a given word and see if it is Palindrome
<CSCI 1100><Mihyar Al-Masalma><B00759975>*/
import java.util.Scanner; // import Scanner class
public class Q2 {
    public static void main(String[] args) {
        String result="";
        // Create an instance of the Scanner class
        Scanner input = new Scanner(System.in);
        // Ask the user to enter the word
        System.out.print("Enter a word: ");
        // Store the word in a variable
        String word = input.nextLine();
        // Convert the string to array of characters
        char[] array = word.toCharArray();
        // iterate over the array starting from the end
        for (int i = array.length-1; i>=0 ;i-- ) {
            // concat characters and assign it to result
            result += array[i];
        }
        // if the words are teh same then print this out
        if (word.equals(result)) {
            System.out.print(word + " is a Palindrome!");
        }else{
            // if not equal print this out
            System.out.print(word + " is NOT a Palindrome!");
        }
    }
}

```

```

> [ ----jGRASP exec: java Q2
> Enter a word: tent
> tent is NOT a Palindrome!
> [ ----jGRASP: operation complete.
> [ ----jGRASP exec: java Q2
> Enter a word: jeje
> jeje is NOT a Palindrome!
> [ ----jGRASP: operation complete.
> [ ----jGRASP exec: java Q2
> Enter a word: racecar
> racecar is 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

```

[ ----jGRASP exec: java Assign2_equals
Enter two words: abccd ceccaa
Common Letters are : ac
[ ----jGRASP: operation complete.

```

//shows when two words have no letters in common

```

[ ----jGRASP exec: java Assign2_equals
Enter two words: ghij abbcc
No Common Letters!
[ ----jGRASP: operation complete.

```

/* Assignment2 Question 3

This program will give the common letters between two words without repetition

<CSCI 1100><Mihyar Al-Masalma><B00759975>*/

import java.util.Scanner; // import Scanner class

public class Q3{

```

    public static void main(String[] args) {
        String result = "";
        // Create an instance of Scanner class
        Scanner input = new Scanner(System.in);
        // Ask the user to enter two words
        System.out.print("Enter two words: ");
        // Store the words in two different variables
        String firstWord = input.next();
        String secondWord = input.nextLine();
        // Assume the first word to be longer
        int length = firstWord.length();
        // Convert it to array
        char[] array = firstWord.toCharArray();
        // Assign secondWord to other
        String other = secondWord;
    }
}

```

```

// if our Assumption is wrong
if (secondWord.length() > length) {
    // length is now the second word length
    length = secondWord.length();
    // Covert second word to array
    array = secondWord.toCharArray();
    // Assign first word to other
    other = firstWord;
}
// iterate starting from 0 to the longer word
for (int i = 0; i < length; i++) {
    // if the letter is in the other word
    if (other.indexOf(array[i]) > -1) {
        // if it is not repeated
        if (result.indexOf(array[i]) == -1) {
            // Add it to the result
            result += array[i];
        }
    }
}
// if there is no common letters
if (result.equals("")) {
    // Print out no common letters
    System.out.print("No Common Letters!");
} else {
    // if there are common letter print it out
    System.out.print("Common Letters are :"+result);
}
}
}

```

```

>> [ ----jGRASP exec: java Q3
    Enter two words: Anna Jackson
    Common Letters are :an
    ----jGRASP: operation complete.

```

```

>> [ ----jGRASP exec: java Q3
    Enter two words: abcd dabc
    Common Letters are :dabc
    ----jGRASP: operation complete.

```

```

>> [ ----jGRASP exec: java Q3
    Enter two words: aaa bbb
    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

```
----jGRASP exec: java Assign2_Shoes
Enter the number of pairs of shoes: 3
Enter the price of pair1 : $26.90
Enter the price of pair2 : $59.99
Enter the price of pair3 : $47.80

Where you would like to ship your shoes:
Nova Scotia, Canada or Other: Nova Scotia

Total Cost for all Shoes: $134.69
Shipping Fee on Order Amount: $20.20
Shipping Fee on Destination: $0.00
TOTAL COST OF ORDER: $154.89
----jGRASP: operation complete.
```

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

```
----jGRASP exec: java Assign2_Shoes
Enter the number of pairs of shoes: 2
Enter the price of pair1 : $56.78
Enter the price of pair2 : $90.0

Where you would like to ship your shoes:
Nova Scotia, Canada or Other: Cadna
That is an incorrect location. Please try again!

Where you would like to ship your shoes:
Nova Scotia, Canada or Other: Canada

Total Cost for all Shoes: $146.78
Shipping Fee on Order Amount: $22.02
Shipping Fee on Destination: $25.00
TOTAL COST OF ORDER: $193.80
----jGRASP: operation complete.
```

```

/* Assignment2 Question 4
This program will take a number of goodies then
calculate how much shipping you should pay and the total
<CSCI 1100><Mihyar Al-Masalma><B00759975>*/
import java.util.Scanner; // import Scanner class
public class Q4 {
    public static void main(String[] args) {
        double price = 0, fees = 0, shippingFees = 0, total = 0;
        String shipping = ""; int counter = 1;
        // Create an instance of the Scanner class
        Scanner input = new Scanner(System.in);
        // Ask the user to enter a number and save it
        System.out.print("Enter the number of pairs of shoes: ");
        int number = input.nextInt();
        // Iterate to get the prices and Store it in a variable
        while (number >= counter) {
            System.out.printf("Enter the price of pair%d : $", counter);
            price += input.nextDouble();
            counter++;
        }
        // Consume the rest of the line
        input.nextLine();
        // Iterate till the user enter valid Shipping destination
        while (!shipping.equals("Nova Scotia") && !shipping.equals("Canada")
&& !shipping.equals("Other")) {
            System.out.println("Where you would like to ship your shoes:");
            System.out.print("Nova Scotia, Canada or Other: ");
            shipping = input.nextLine();
        }
        // Print out in formatted way the cost of the shoes
        System.out.printf("Total Cost for all Shoes: $ %.2f", price);
        // Based on the price, calculate the shipping fees
        if (price < 100) {
            fees = price * 0.25;
        } else if (price < 200) {
            fees = price * 0.15;
        } else if (price > 200) {
            fees = price * 0.10;
        }
        // Print out the shipping fees
        System.out.printf("\nShipping Fee on Order Amount: $ %.2f", fees);
        // based on the destination calculate the fees
        if (shipping.equals("Canada")) {
            shippingFees = 25;
        } else if (shipping.equals("Other")) {
            shippingFees = 50;
        }
        // print out the result in a formatted way
        System.out.printf("\nShipping Fee on Destination: $ %.2f",
shippingFees);
        total = price + fees + shippingFees;

        System.out.printf("\nTOTAL COST OF ORDER: $ %.2f", total);
    }
}

```

```
----jGRASP exec: java Q4
>> Enter the number of pairs of shoes: 3
>> Enter the price of pair1 : $25.51
>> Enter the price of pair2 : $26.32
>> Enter the price of pair3 : $55.12
Where you would like to ship your shoes:
>> Nova Scotia, Canada or Other: Nova Scotia
Total Cost for all Shoes: $ 106.95
Shipping Fee on Order Amount: $ 16.04
Shipping Fee on Destination: $ 0.00
TOTAL COST OF ORDER: $ 122.99
----jGRASP: operation complete.
```

```
----jGRASP exec: java Q4
>> Enter the number of pairs of shoes: 2
>> Enter the price of pair1 : $15.99
>> Enter the price of pair2 : $99.9
Where you would like to ship your shoes:
>> Nova Scotia, Canada or Other: Cadna
Where you would like to ship your shoes:
>> Nova Scotia, Canada or Other: Canada
Total Cost for all Shoes: $ 115.89
Shipping Fee on Order Amount: $ 17.38
Shipping Fee on Destination: $ 25.00
TOTAL COST OF ORDER: $ 158.27
----jGRASP: operation complete.
```

```
----jGRASP exec: java Q4
>> Enter the number of pairs of shoes: 4
>> Enter the price of pair1 : $11.9
>> Enter the price of pair2 : $9.11
>> Enter the price of pair3 : $14.5
>> Enter the price of pair4 : $15.5
Where you would like to ship your shoes:
>> Nova Scotia, Canada or Other: Other
Total Cost for all Shoes: $ 51.01
Shipping Fee on Order Amount: $ 12.75
Shipping Fee on Destination: $ 50.00
TOTAL COST OF ORDER: $ 113.76
----jGRASP: operation complete.
```


Method printf (Textbook pp. 164-175)

- Performs formatted output
- `System.out.printf(formatString, argumentList)`
- `formatString` is a string that contains text and/or formatting specifiers
- argument list is a list of zero or more additional arguments which are formatted to the specifiers listed in the `formatString`
- See the textbook for all the formats for output (e.g., `%d` is replaced with an int, `%s` is replaced with a String)

Examples:

```
int currYr=2016;  
System.out.printf("The current year is %d", currYr); // %d tells java to print an int  
// this replaces the format "%d" with the int currYr
```

Prints: The current year is 2016

You can have more than one argument:

```
int age1=6;  
int age2=8;  
String name1="Sally";  
String name2="Timmy";  
System.out.printf("%s is %d and %s is %d years old.", name1, age1, name2, age2); // %s is a string  
// matches the first %s with the first argument after the comma (name1)
```

Prints: Sally is 6 and Timmy is 8 years old.

You can format decimal places

```
double ave = 3.5678;
```

```
System.out.printf("The average is %.2f", ave); // %.2f is a decimal with 2 decimal places (rounded)
```

Prints: The average is 3.57