**King Saud University**

**College of Computer & Information Science**

**CSC111 – Lab03**

**IO, Variables, Expressions**

**All Sections**

**-------------------------------------------------------------------**

# Instructions

Web-CAT submission URL:

http://10.131.240.28:8080/Web-CAT/WebObjects/Web-CAT.woa/wa/assignments/eclipse

# Objectives:

1. Student should learn how to read a problem statement and analyze it as following:
   1. Find out if program needs input, how many inputs it is going to accept and of what type.
   2. Decide if variables are needed, how many variable and of what type.
   3. Understand the computation operations that are needed to solve the problem and design a simple algorithm to solve it (i.e., if program needs to compute certain values using arithmetic expression).
   4. Decide what is the program is going to output to the end user.
2. Student should learn how to define variable, and assign them values.
3. Students should learn how to write arithmetic expressions and use operators.
4. Students should learn about different numeric data types.

# Lab Exercise 1

Write a program that prompts the user to enter two points (x1, y1) and (x2, y2) and displays their distance between them. The formula for computing the distance is .

Note that you can use Math.pow(a, 0.5) to compute .

Here is a sample run:

Enter x1 and y1: 1.5 -3.4 **↵**

Enter x2 and y2: 4 5 **↵**

The distance between the two points is 8.764131445842194

# Solution

1. Create a new project in eclipse and name it **lab03**
2. Create a new class and name it **Distance**. Make sure you choose the public static void main option.
3. Write the program as following (you can ignore comments):

**import** java.util.Scanner;

**public** **class** Distance {

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.***in***);

// Enter the first point with two double values

System.***out***.print("Enter x1 and y1: ");

**double** x1 = input.nextDouble();

**double** y1 = input.nextDouble();

// Enter the second point with two double values

System.***out***.print("Enter x2 and y2: ");

**double** x2 = input.nextDouble();

**double** y2 = input.nextDouble();

// Compute the distance

**double** distance = Math.*pow*((x1 - x2) \* (x1 - x2) +

(y1 - y2) \* (y1 - y2), 0.5);

System.***out***.println("The distance between the two points is " + distance);

}

}

1. When you are done, save your program and run it. Make sure it prints the output as shown above.
2. Submit your program to WebCAT through eclipse to get familiar with WebCAT. Ask your TA for help.

# Lab Exercise 2

# Write a program that reads an integer between 0 and 1000 and adds all the digits in the integer. For example, if an integer is 932, the sum of all its digits is 14.

# Hint: Use the % operator to extract digits, and use the / operator to remove the extracted digit. For instance, 932 % 10 = 2 and 932 / 10 = 93.

Here is a sample run:

Enter an integer between 0 and 1000: 999 **↵**

The sum of all digits in 999 is 27

# Solution

1. Use the same project **lab03** that you created before
2. Create a new class and name it **SumDigits**. Make sure you choose the public static void main option.
3. Write the program as following (you can ignore comments):

// Summarize all digits in an integer < 1000

**public** **class** SumDigits {

// Main method

**public** **static** **void** main(String[] args) {

java.util.Scanner input = **new** java.util.Scanner(System.***in***);

// Read a number

System.***out***.print("Enter an integer between 0 and 1000: ");

**int** number = input.nextInt();

// Find all digits in number

**int** lastDigit = number % 10;

**int** remainingNumber = number / 10;

**int** secondLastDigit = remainingNumber % 10;

remainingNumber = remainingNumber / 10;

**int** thirdLastDigit = remainingNumber % 10;

// Obtain the sum of all digits

**int** sum = lastDigit + secondLastDigit + thirdLastDigit;

// Display results

System.***out***.println("The sum of all digits in " + number

+ " is " + sum);

}

}

1. When you are done, save your program and run it. Make sure it prints the output as shown above.
2. Submit your program to WebCAT through eclipse to get familiar with WebCAT. Ask your TA for help.

**Done…**