**COSC 1436: Programming Fundamentals I  
Lab 8: while and do..while loops**

Please submit one source code file (.java file) for each exercise. (Ex. Lab5Ex1.java, Lab5Ex2,java, Lab5Ex3.java ...etc).

Remember to follow the commenting guidelines.

**Program 1: Unit conversion –** while **version (20 points):**

In this program, you will ask the user for a number (double) of inches and convert that number into centimeters. Remember that 1 inch = 2.54 cm. After you print out the answer, you will ask the user if they want to enter another number. If the user answer ‘Y’ or ‘y’, loop around and ask the user for another number of inches. If the user answers anything other than ‘Y’ or ‘y’, print “Goodbye” and quit the program.

You must use a while loop in this program to handle whether the user wants to enter another number.

Sample output:

Enter the number of inches.

10.0

10.0 inches is 25.4 cm.

Do you want to enter another number?

Y

Enter the number of inches.

2.0

2.0 inches is 5.08 cm.

Do you want to enter another number?

N

Goodbye

Tip: Java can be weird sometimes when reading from the keyboard, especially about the Return (other known as the Enter, or new line) character. If your program always gives you a StringIndexOutOfBoundsException when you run it, try adding the following line right after you print “Do you want to enter another number?”

String extra\_newline=keyboard.nextLine( );

That should take care of any Return characters that Java has problems with.

import java.util.Scanner;

public class Lab08Prog1{

public static void main(String[] args){

Scanner keyboard=new Scanner(System.in);

double inches=1;

char again='Y';

String extra\_newline;

again='Y';

while( (again=='Y') || (again=='y')){

System.out.println("Enter the number of inches.

inches=keyboard.nextDouble( );

System.out.println(inches+ " is "+ 2.54\*inches + " cm”);

System.out.println("Do you want to enter another number?”);

extra\_newline=keyboard.nextLine();

again=keyboard.nextLine().charAt(0);

}

System.out.println("Goodbye”);

}

}

**Program 2: Unit conversion –** do..while **version (20 points):**

Rewrite Program 1 using a do..while loop instead of a while. You can assume that the user will want to enter at least one value (in other words, no 0 iteration loops).

import java.util.Scanner;

public class Lab08Prog2{

public static void main(String[] args){

Scanner keyboard=new Scanner(System.in);

double inches=1;

char again='Y';

String extra\_newline;

do{

System.out.println("Enter the number of inches”);

inches=keyboard.nextDouble();

System.out.println(inches+ " is "+ 2.54\*inches + " cm”);

System.out.println("Do you want to enter another number?**”);**

extra\_newline=keyboard.nextLine();

again=keyboard.nextLine().charAt(0);

} while( (again=='Y') || (again=='y'));

System.out.println("Goodbye”);

}

}

Program 3: Factorization – **do..while** version (20 points):

For any positive integer, you can compute how many numbers evenly divide into it. For instance, the number 12 has the following factors:

* 12/1 = 12
* 12/2 = 6
* 12/3 = 4
* 12/4 = 3
* 12/6 = 2
* 12/12 = 1

The factors of 12 are 1, 2, 3, 4, 6, and 12. None of those numbers will have a remainder when 12 you divide 12 by them. It turns out that computing all the factors of a number is a surprisingly useful operation in computer science. A very popular cryptography scheme relies on the fact that calculating the factors of very large numbers can take a lot of time.

Write a program that asks the user for a positive integer (which we will call *n*). It will then loop through all number from 1 to *n* to see if it can be divided without a remainder. If it can, your program will print it out.

Sample output:

Which number do you want to factor? 18

The factors of 18 are:

1

2

3

6

9

18

import java.util.Scanner;

public class Lab07Prog3{

public static void main(String[] args){

Scanner keyboard=new Scanner(System.in);

int factor=1;

int n=0;

System.out.print("Which number do you want to factor?

n=keyboard.nextInt();

System.out.println("The factors of "+ n + " are

factor=1;

while( factor<=n){

if (n%factor == 0){

System.out.println(factor);

}

factor++;

}

}

}

**Program 4: Factorization –** while **version (20 points):**

Rewrite Program 3 using a while loop instead of a do..while.

import java.util.Scanner;

public class Lab07Prog4{

public static void main(String[] args){

//import java.util.Random;

Scanner keyboard=new Scanner(System.in);

int factor=1;

int n=0;

System.out.print("Which number do you want to factor?

n=keyboard.nextInt();

System.out.println("The factors of "+ n + " are ");

factor=1;

do{

if (n%factor == 0){

System.out.println(factor);

}

factor++;

} while( factor<=n);

}

}

**Extra Credit Program (10 points)**:

Program 3 can run much faster, especially for large values for *n*. It turns out that you do not need to loop through all values from 1 to *n* to compute all of the factors. Optimize Program 3 by figuring out the minimum number of times you really need to go through the loop.

Give 5 points if instead of n, they loop to ½ n.

Give 10 points if they loop to

## Multiple Choice Questions (20 points)

1) Which loop is a post-test loop? There may be more than one.

a. for loop

b. while loop

c. do..while loop

2) Which loop(s) are pre-test loops? There may be more than one.

a. for loop

b. while loop

c. do..while loop

3) When should you use a for loop over a while or do..while loop?

a. When you know how many times you want a certain piece of code to run.

b. When you need to traverse arrays.

c. When you need to nest a loop.

4) When should you use a do..while loop over a for loop or while loop?

a. When you know how many times you want a certain piece of code to run.

b. When you want a piece of code to run at *least* once.

c. When you want to test a condition first before running.

5) You are writing a program that uses a loop to completely erase a computer by deleting every file, one file at a time. Before deleting a file, your program asks the user “Are you sure you want to continue? Y/N”. You do not know how many files are on your computer and you want the user to be able to exit before deleting anything. What kind of loop should you use?

a. for loop

b. while loop

c. do..while loop