GJU

الجامعة الألمانية الأردنية
German Jordanian University

CS1160
Lab Assignment 4: Loops (for, while, do…while)

Date: 01 November 2020 **Online Lab Session**

I. Overview

In this lab, you will learn about for, while, and do…while. You will also learn how to use them in real life applications.

There are a basic concepts that are related to the loops:

1- Initial expression: an expression that creates the starting point of the loop.
2- Loop condition: an expression that must be true to keep executing the loop whatever inside the loop scope.
3- Loop expression update (incremental) : an expression that keeps updating the value of the loop variable.

II. Review

1- for loop:

General expression :

```
for (initial_expression; loop_condition; loop_expression ){

        statements;

}
```

2- while loop:

General Expression:

```
Initial_expression;
while (condition_expression)

{
            Statement 1;
                     :
            incremental_expression;

}
```

3- do … while loop:

General Expression:

```
        initial_expression;

        do {




        incremental_expression;

        } while(condition_expression);
```

III. Operators

An operator is a symbol that tells the compiler to perform specific mathematical or logical functions.

- Loops arithmetic related operators:

Operator	How it is written	Example	Value in Memory Before Using	Output	Value in Memory After Using

post-increment	varName++	int x = 3;
		printf(“x = %d\n“,x++)
			X = 3	3	4

pre-increment	++varName	int y = 5;
		printf(“y = %d\n“,++y)
			Y = 5	6	6

post-decrement	varName--	int w = 9;
		printf(“w = %d\n“,w--)
			W = 9	9	10

pre-decrement	--varName	int z = 15;
		printf(“z = %d\n“,--z)
			Z = 15	14	14

III. Tasks

1- Write a C program to print the numbers from 1 to 20 (using loops).

2- Write a C program that reads from the user an integer number **n** and prints out the numbers from 1 to **n**.

3- Write a C program to print the even integer numbers from 1 to **n,** knowing that **n** is an integer entered by the user.

4- Write a C program that print the sum of the numbers from **m** to **n,** knowing that **m and n** are integers entered by the user.

5- Write a C program that calculates and print the sum of the odd numbers from **1** to **n,** knowing that **n** are integers entered by the user and then calculates and prints the average of them.

6- Write a C program to read an integer number **num** and find it's factorial.

Note: the factorial number is all the number multiplied from 1 to the number itself.

$$n! = n(n-1)(n-2)\dots(2)(1)$$

Example :

5! = (5)(4)(3)(2)(1) = 120

7- Write a C program that reads a number and check whether its' prime or not.

Note: a prime number is the number that can only be divisible by 1 and it's self.

Examples: 2, 3, 5, 7, 11, 13… etc.

8- Write a C program to print Fibonacci series up to n terms using loop.

Note: *Fibonacci series* is a series of numbers where the current number is the sum of previous two terms.

Example: 0, 1, 1, 2, 3, 5, 8, 13, 21, ... , (n-1th + n-2th)