Programming Fundamentals
(Lab 2019)

Lab Manual

Computer Science for Session 2019 (Fall-2019)

Programing Fundamentals

(Lab 6)

Target: For loops, do while, while loops and nested control structures.

CLOs to be checked: Plagiarism, Code commenting & indentation, Logic Building

i)   Guidelines/Instructions:
- This is an important. Invest your time on all problems. Solve it with your own logic. You might get solutions of some questions on internet, but it will not help you later.
- Discussion with friends is fine but watching other's code or showing your code to others is plagiarism.
- More practice => more polished brain => Confidence => Alive conscience.
- Less practice => less confidence => More fear => Leads to plagiarism => Failure.
- In this lab, plagiarism CLO, code commenting & indentation and logic building will be evaluated. Make sure you follow all the rules.

ii)   Reading Material:
- **Consult Chapter No 5 of book "C++: Programming from Problem Analysis to Program Design by D.S Malik (latest edition)" for better understanding of given problems.**

iii)   Tasks to be completed in lab:

Example program 1:

```
//********************************************************************
Nested For loop(write C++ Program to Print these stars)
//********************************************************************
        *
        **
        ***
        ****
        *****
```

Syntax Nested For Loop

```
for (initialization; condition; increment/decrement)
{
   statement(s);
   for (initialization; condition; increment/decrement)
   {
     statement(s);
     ... ... ...
   }
   ... ... ...
}
```

Instructor: Ms. Sahar Waqar
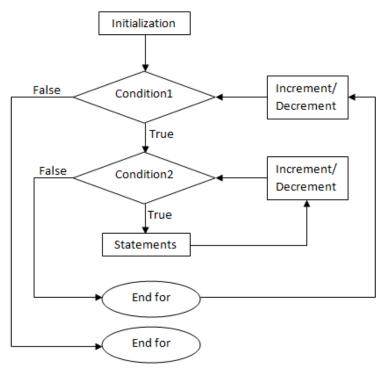**Graduate Assistant: Mr. Muhammad Nazam Maqbool, Mr. Usman Ghani**

Fig: Flowchart for nested for loop

Clearly, you want to print five lines of stars. In the first line, you want to print one star, in the second line, two stars, and so on. Because five lines will be printed, start with the following for statement:

for (i = 1; i <= 5; i++)

The value of i in the first iteration is 1, in the second iteration it is 2, and so on. You can use the value of I as the limiting condition in another for loop nested within this loop to control the number of stars in a line.

```cpp
#include<iostream>
using namespace std;
int main()
{
    for (i = 1; i <= 5; i++)        //Line 1
    {                               //Line 2
        for (j = 1; j <= i; j++)    //Line 3
            cout << "*";            //Line 4
        cout << endl;               //Line 5
    }

    return 0;
}                                   //Line 6
```

A walk-through of this code shows that the for loop in Line 1 starts with i = 1. When i is 1, the inner for loop in Line 3 outputs one star and the insertion point moves to the next line. Then i becomes 2, the inner for loop outputs two stars, and the output statement in Line 5 moves the insertion point to the next line, and so on. This process continues until i becomes 6 and the loop stops.

Example program 2:
//**

Nested While Loop(write C++ program to print the number pattern.)
//***

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Syntax
```
while (condition1)
{
   statement(s);
   while (condition2)
   {
      statement(s);
      ... ... ...
   }
   ... ... ...
}
```
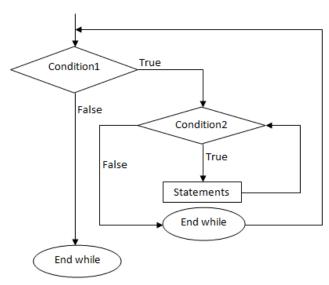

fig: Flowchart for nested while loop

Code:

```cpp
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int i=1,j;
    while (i <= 5)
    {
        j=1;
        while (j <= i )
        {
            cout <<j;
            j++;
        }
        cout << endl;
        i++;
    }
    getch();
    return 0;
}
```

Example program 3:

//***

Nested do While Loop(write C++ program to print the stars)

//***

*
**
***
****
*****

Syntax of Nested do while loop

```
do
{
  statement(s);
  do
  {
    statement(s);
    … … …
  }while (condition2);
  … … …
}while (condition1);
```
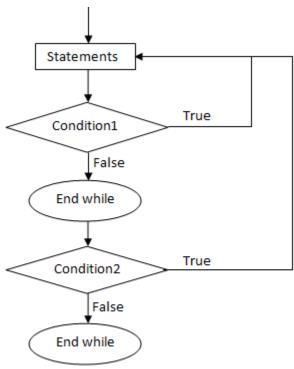
Fig: Flowchart for nested do-while loop

```cpp
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int i=1,j;
    do
    {
        j=1;
        do
        {
            cout << "*";
            j++;
        }while(j <= i);
        i++;
        cout << endl;
    }while(i <= 5);

    getch();
    return 0;
}
```

iv)   **Tasks to be completed in lab:**

1. Write a C++ program which asks user to enter two numbers: a dividend (the top number in a division) and a divisor (the bottom number). It then calculates the quotient (the answer) and the remainder, using the / and % operators, and prints out the result also ask user to try again.(if user enter 'y' or 'Y' then program again ask for two number if user enter 'n' or 'N' then it terminate). Solve this problem using **for loop control structure.**

   Expected Output:

   ```
   Enter dividend: 11
   Enter divisor: 3
   Quotient is 3, remainder is 2
   Do another? (y/n): y
   Enter dividend: 222
   Enter divisor: 17
   Quotient is 13, remainder is 1
   Do another? (y/n): n
   ```

2. Write a C++ for the following problem. Assume that you want to generate a table of multiples of any given number. Write a program that allows the user to enter the number and then generates the table, formatting it into **n** columns and **m** lines. For n = 10 and m = 20, interaction with the program should look like this (only the first three lines are shown):

   ```
   Enter a number: 7
      7   14   21   28   35   42   49   56   63   70
     77   84   91   98  105  112  119  126  133  140
    147  154  161  168  175  182  189  196  203  210
   ```

3. Write a C++ program that calculates value of **e** using following series.

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} \ldots$$

   User will provide **n** i.e. number of terms of series and **precision** (number of digits after decimal places) and **e** will be the value of sum of n terms of the series. For example, if user enters 0, output of the program is 0. If user enters 1 as **n** and 1 as **precision**, output is 1.0. If user enters 2 as **n** and 0 as **precision**, output is (1+1/1!) 2. If user enters 5 as **n** and 4 as **precision**, output of the program is (1+1/1! + 1/2! + 1/3! + 1/4) 2.7083.

4. Write a C++ program which calculates Fibonacci series and prints the series up to **n** terms. Each term (other than 1st two) in Fibonacci series is calculated by the sum of previous two terms. For example:

   If n = 0, output is None.
   If n = 1, output is 1.
   If n = 2, output is 2.
   If n = 3, output is 1, 2, 3
   If n = 5, output is 1, 2, 3, 5, 8
   If n = 10, output is 1, 2, 3, 5, 8, 13, 21, 34, 55, 89
   And so on.

5. Write a C program to find power of a number using while loop.

v)   **Tasks to be submitted:**

Instructor: Ms. Sahar Waqar
Graduate Assistant: Mr. Muhammad Nazam Maqbool, Mr. Usman Ghani

1. Generate the following output using nested for loops in C++. User should input number of **rows** and number of **columns** and you will print the table starting from 1 up to rows. Make sure you adjust the formatting before submitting the assignment.

```
1   2   3   4   5   6   7   8   9  10
2   4   6   8  10  12  14  16  18  20
3   6   9  12  15  18  21  24  27  30
4   8  12  16  20  24  28  32  36  40
5  10  15  20  25  30  35  40  45  50
```

2. Write a C++ program which takes number of students ''nOfStudents' and total number of subjects **nSubjects**' as input. Program then asks each student to enter his marks for **nSubjects**. When student enters -99, next students enters his marks. When any student enters -100 or causes input failure, program stops the execution abruptly and prints a message "Program ended abruptly. Do you want to continue (Y/N) ?". If user enters Y, program restarts the execution and asks nOf Students and nSubjects again and so on. If user enters N, program stops the execution. If no input failure occurs and no student enters -100, then average marks for all students will be displaced as output. Also, note that you have to validate that each student must enter marks for exactly **nSubjects** before entering -999 otherwise your average result will not be correct. If user enters, -999 before nSubjects terms then you have re-ask him to enter the marks.
For example,
If user enters nOfStudents = 3 and nSubjects = 4 then possible output of the program is,
Please enter number of Students = 3
Please enter number of Subjects = 4
Student1: 98 87 80 -999
Student2: 80 55 -999
You have entered marks for 3 subjects. Please re-enter:
Student2: 80 55 -100 87 -999
Program ended abruptly. Do you want to continue (Y/N)?
Y
Please enter number of Students = 3
Please enter number of Subjects = 4
Student1: 98 87 80 -999
Student2: 80 55 40 -999
Student3: 70 87 65 -999
****
Average of your class marks = 220.67 (up to 2 decimal places)

3. Use for loops to construct a program that displays a pyramid of Xs on the screen. Take number of rows as input. The pyramid should look like this
```
    X
   XXX
  XXXXX
 XXXXXXX
XXXXXXXXX
```

4. The population of a town A is less than the population of town B. However, the population of town A is growing faster than the population of town B. Write a program that prompts the user to enter the population and growth rate of each town. The program outputs after how many years the population of town A will be greater than or equal to the population of town B and the populations

of both the towns at that time. (A sample input is: Population of town A = 5000, growth rate of town A = 4%, population of town B = 8000, and growth rate of town B = 2%.)

5. Suppose you give a dinner party for six guests, but your table seats only four. In how many ways can four of the six guests arrange themselves at the table? Any of the six guests can sit in the first chair. Any of the remaining five can sit in the second chair. Any of the remaining four can sit in the third chair, and any of the remaining three can sit in the fourth chair. (The last two will have to stand.) So, the number of possible arrangements of six guests in four chairs is 6*5*4*3, which is 360. Write a program that calculates the number of possible arrangements for any number of guests and any number of chairs. (Assume there will never be fewer guests than chairs.) Don't let this get too complicated. A simple for loop should do it.

6. Write C++ code **power-close-to** that takes as arguments two positive integers b and n, and returns the smallest power of b that is greater than n. That is, it should return the smallest positive integer e such that $b^e > n$.

7. Write a C++ program to print all **perfect** numbers between 1 to n. For example, 6 is perfect number since divisor of 6 are 1, 2 and 3. Sum of its divisor is $1 + 2 + 3 = 6$ and 28 is also a perfect number since $1 + 2 + 4 + 7 + 14 = 28$. Other perfect numbers: 496, 8128

8. Write a C++ program to check whether a number is **strong** number or not. *Strong number* is a special number whose sum of factorial of digits is equal to the original number. For example: 145 is strong number. Since, $1! + 4! + 5! = 145$

9. Write a C++ program which tells if a number is a palindrome or not. Palindrome is a sequence of characters which when read from left to right or right to left remains the same. For example, a 121 is a palindrome. 45654, 22 and 9983899 etc. are also palindromes. 1231 is not palindrome.

10. Write a C++ program which takes an integer as input and prints that number in words. For example, if user enters 450 then output will be "Four hundred and fifty". If user enters 53, then output of the program is "Fifty-three". If user enters 98754 then output of the program is Ninety-eight thousand seven hundred and fifty-four. Maximum limit for input is 5 digits.

11. There are 11 hard disks. One of them is used for backup and rest for saving one digit. One hard disk gets corrupted. How will you retrieve loss data? Write a C++ program for this problem. Ask about values of all the hard disks and use back up hard disk as storage for your algorithm. Ask user again about the number of one hard disk which gets corrupted and save -1 in it. Using your algorithm and recover the corrupted data. You should not duplicate data of your hard disks in any other variable. You should focus on the question "how can you recover the data using your algorithm"

vi) **Submission instructions:**
- Submit all questions with name q1.cpp, q2.cpp and qn.cpp so on.

vii) **Office hours and email address for communication:**

- Ms. Sahar Waqar (Teacher)
  - **Email:** sahar.waqar@uet.edu.pk
  - **Office hours:**
    - Tuesday (**Fixed**): 11:00 pm to 12:45 pm
    - Wednesday (Variable): 11:00 am to 1:00 pm (May change due to meetings)
    - Wednesday (**Fixed**): 1:00 pm to 3:00 pm
- Mr. Usman Ghani (Graduate assistant)
  - **Email**:engrusmaanghani@gmail.com
- Mr. Muhammad Nazam Maqbool (Graduate assistant)
  - **Email**: ashirmaqbool1611@gmail.com

Instructor: Ms. Sahar Waqar
Graduate Assistant: Mr. Muhammad Nazam Maqbool, Mr. Usman Ghani

viii)   Late submissions:

Late submissions are accepted with certain fraction of marks deducted. Email your late submissions at assignmentsuet@gmail.com. You cannot delay any lab, longer than one week.