Sukkur IBA University

Faculty of Computer Science

Fundamentals of Programming (Fall-2019)

Lab 06

Instructor: Saif Hassan **Date: 7th Sep, 2019**

Objective of Lab No. 06:

After performing lab 5, students will be able to:

- understand pre/post increment/decrement
- use loops to design repetitive tasks

Practice Task:

Write a program to print and sum 1 – 10 numbers.

Solution

FOR LOOP

```
#include <iostream>
using namespace std;
int main()
{
    int i;
    int sum = 0;
    for (i = 1; i <= 10; i++)
    {
        cout << i << " ";
        sum = sum + i;
    }
    cout << endl;
    cout << endl << sum;

    return 0;
}
```

WHILE LOOP

```
#include <iostream>
using namespace std;
int main(){
    int i=1;
    int sum = 0;
    while(i<=10)
    {
       cout<<i<<" ";
       sum = sum + I;
       i++;
    }
    cout<<endl<<sum;

    return 0;

}
```

DO-WHILE LOOP

```
#include <iostream>
using namespace std;
int main(){
    int i=1;
    int sum = 0;
    do
    {
       cout<<i<<" ";
       sum = sum + I;
       i++;
    }
    while(i<=10);
    cout<<endl<<sum;

    return 0;
}
```

Exercises

Refreshing Task01 (PRE/POST INCREMENT)

Code is provided below, perform following tasks:

a) Is there any syntax error in code? (Mention error names and line number).
b) Is there any logical error in code? (Mention error names and line number).
c) Resolve both errors on paper and observe the output.
d) Execute code on Machine/System and match the output, you noted/observed.

<u>Note: Teacher can ask anyone to explain your observation.</u>

```
1.  cout<<x<<", "<<y<<endl;
2.
3.  int x, y;
4.  cout<<x<<", "<<y<<endl;
5.
6.  x = 5, y = 9;
7.  cout<<x<<", "<<y<<endl;
8.
9.  cout<<x++<<", "<<y<<endl;
10. cout<<"x"<<", "<<y<<endl;
11. cout<<++x<<", "<<y++<<endl;
12.
13. ++x;
14. y = 6;
15. cout<<++x<<", "<<y++<<endl;
16. cout<<x++<<", "<<++y<<endl;
17.
18. x = x + 1 // x++ or x += 1
19. y += 2 // y = y + 2
20.
21. cout<<++x<<", "<<y++<<endl;
22. cout<<x<<", "<<y<<endl;
```

Refreshing Task02 (PRE/POST DECREMENT)

Code is provided below, perform following tasks:

a) Is there any syntax error in code? (Mention error names and line number).
b) Is there any logical error in code? (Mention error names and line number).
c) Resolve both errors on paper and observe the output.
d) Execute code on Machine/System and match the output, you noted/observed.

<u>Note: Teacher can ask anyone to explain your observation.</u>

```
1.  cout<<x<<", "<<y<<endl;
2.
3.  int x, y;
4.  cout<<x<<", "<<y<<endl;
5.
```

```
x = 5, y = 9;
cout<<x<<", "<<y<<endl;

cout<<x--<<", "<<y<<endl;
cout<<--y<<", "<<--y<<endl;
cout<<--x<<", "<<y--<<endl;

--x;
y = 6;
cout<<y--<<", "<<x<<endl;
cout<<++y<<", "<<--y<<endl;

x = x - 1 // x-- or x -= 1
y -= 2 // y = y - 2

cout<<y++<<", "<<"x"<<endl;
cout<<x<<", "<<y<<endl;
```

Task01 (OBSERVING FOR/WHILE/DO-WHILE BEHAVIOR)

Write a program to ask user input **START/END** and iterate loop from start to end and perform following tasks:

a) Print numbers using **FOR/WHILE/DO WHILE**.

b) Test on following values and observe output (You may be asked anytime during lab regarding this observation).

Start	End	Observation
1	10	
5	5	
1	-10	
10	1	

c) While testing on last two inputs, it may work abnormally. Now modify above program so that it should work **t** time where **t** is difference between start/end.

d) How for/while/do-while loop behaves?

Task02 (GRADE CALCULATION)

Write a program to ask user input **n_courses** (Total Courses), then iterate loop **n_courses** time and ask user **marks** in each course and perform following tasks:

a) Calculate **average** of all courses marks.

b) Calculate total **percentage** of all courses.

c) Tell the user whether s/he is **PASS/FAIL, FAIL** → less than **60**, and otherwise **PASS**.

Task03

Write a program to output the **N** terms of **HARMONIC** series and their **SUM**.

```
Input:     5
Output:    1/1 + 1/2 + 1/3 + 1/4 + 1/5
           Total Sum is: 2.28333
```

Task04 (CONTINUE/EXIT)

Write a program to ask user **NAME** and print “Welcome Dear NAME”, then ask user whether user wants to continue or exit (y → continue, e → exit). Print appropriate message to tell user how to **EXIT/CONTINUE**.

Note: In following each task, you have to ask user whether **CONTINUE/EXIT**, if continue then program should **RUN AGAIN** otherwise **TERMINATE**.

Task05 (PRIME NUMBERS)

Write a program to ask user input **N** and perform following tasks:

a) Check whether **N** is **PRIME/NOT**?
b) Generate first **N PRIME** numbers.

Note: **PRIME** numbers are those which are **perfectly divided** by **1** and **itself**.

Task06 (FIBONACCI NUMBERS)

Write a program to ask user input **N** and perform following tasks:

a) Check whether **N** comes in **FIBONACCI** series or **NOT**?
b) Generate **FIBONACCI** series till **N.**

Note: **FIBONACCI** series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55…

Task07 (SIMPLE PATTERNS)

Write a program to ask user input **N** and design following patters:

Input N = 5
Output:

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

```
A
A   B
A   B   C
A   B   C   D
A   B   C   D   E
```