National University
of Computer & Emerging Sciences

Programing Fundamentals Lab

FALL - 2022

LAB 03

Learning Outcomes

In this lab you are expected to learn the following:

- Variables and storage
- Stream insertion/extraction operators

Exercise 1: A Simple First Program

You need to perform the following to complete the task.

1. Open the Terminal (Ctrl + Alt + t)
2. Installation g++

   lsudo apt install g++

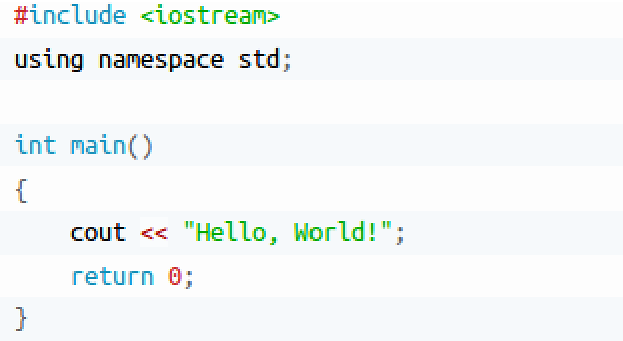
   Note: g++ is already install on lab PCs
3. Create file of .cpp file extension using touch command

   touch helloworld.cpp

4. Now open the text editor using gedit command

   gedit helloworld.cpp

5. Write the following code in helloworld.cpp file.



```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello, World!";
    return 0;
}
```

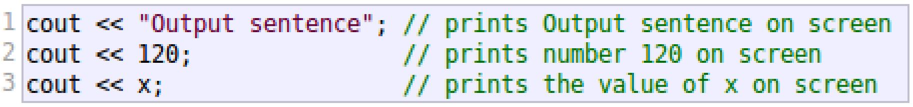
6. Save and close the file.

7. compile and execute it

```
$ g++ -o hello helloworld.cpp

$ ./hello
```

Standard output (cout)

cout is a C++ stream object, used for standard output by default is the screen. For formatted output operations, cout is used together with the *insertion operator*, which is written as <<(i.e., two "less than" signs).



```
cout << "Output sentence"; // prints Output sentence on screen
cout << 120;               // prints number 120 on screen
cout << x;                 // prints the value of x on screen
```

Standard input (cin)

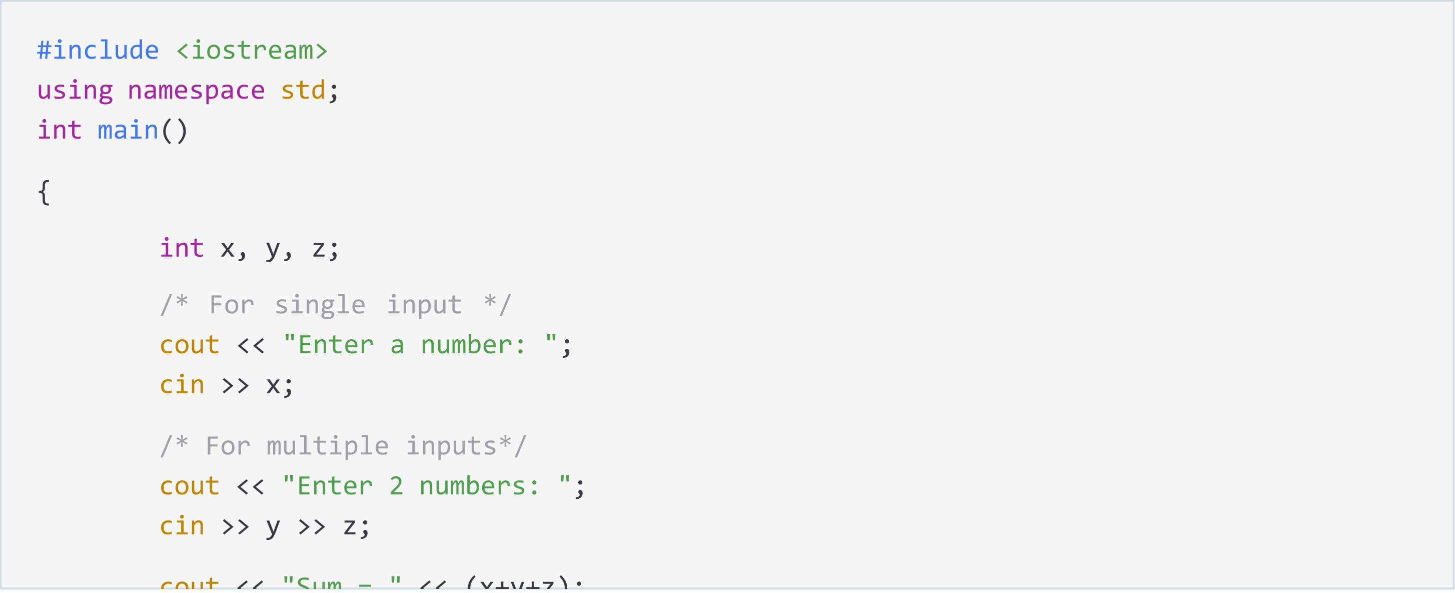
In most program environments, the standard input by default is the keyboard, and the C++ stream object defined to access it is cin.

For formatted input operations, cin is used together with the extraction operator, which is written as

>> (i.e., two "greater than" signs). This operator is then followed by the variable where the extracteddata is stored. For example:

```
int age;
cin >> age;
```

Example: cin with extraction operator:



```
#include <iostream>
using namespace std;
int main()

{

        int x, y, z;

        /* For single input */
        cout << "Enter a number: ";
        cin >> x;

        /* For multiple inputs*/
        cout << "Enter 2 numbers: ";
        cin >> y >> z;

        cout << "Sum = " << (x+y+z);
```

When you run the program, a possible output will be:

```
Enter a number: 9 Enter
2 numbers: 1 5Sum = 15
```

Constants and Variables:

Constants: A specific alphabetical and/or numeric value that is never changed.
For Ex. PI - 3.14159

Variables: The value that can be changed.

For Ex. ShoeCost = 56.00 and ShoeCost = 35.00

Data Types:

1. **int - integer:** a whole number.
This data type is used to define an integer number (-…. -3, -2,-1,0,1,2,3….). A single integer occupies 2 bytes.
For example: int a;
declares that you want to create an int variable called a.
To assign a value to our integer variable we would use the following C statement: a=10;

2. **float** - floating point value: i.e. a number with a fractional part.
A float, or floating point, number has about seven digits of precision and a range of about 1.E-36 to 1.E+36. A float takes four bytes to store.

3. **double -** a double-precision floating point value.
A double, or double precision, number has about 13 digits of precision and a range of about 1.E-303 to 1.E+303. A double takes eight bytes to store.
Note: Single precision and Double precision basically differs in the number of digits represented after the decimal point. Double precision number will represent more digits after the decimal point than a single precision number. Example: Single precision – 32.75 and double precision – 32.7543

4. **char -** a single character.
Used to define characters. A single character occupy 1 byte.
To assign, or store, a character value in a char data type is easy - a character variable is just a symbol enclosed by single quotes.

char a;
char a = ‘10’;

Escape Sequences

Character combinations consisting of a backslash (\) followed by a letter or by a combination of digits are called "escape sequences." To represent a newline character, single quotation mark, or certain other characters in a character constant, you must use escape sequences. An escape sequence is regarded as a single character and is therefore valid as a character constant. Escape sequences are used to format our output. The following escape sequences can be used to print out special characters.

Escape Sequence	**Description**
\n	Newline
\t	Horizontal tab
\\	Backslash

\'	Single quote
\"	Double quote

To insert a line break, a new-line character shall be inserted at the exact position the line should be broken. In C++, a new-line character can be specified as \n (i.e., a backslash character followed by a lowercase n). For example:

```
cout << "First sentence.\n";
cout << "Second sentence.\nThird sentence.";
```

This produces the following output:

```
First sentence.
Second sentence.
Third sentence.
```

Alternatively, the endl manipulator can also be used to break lines. For example:

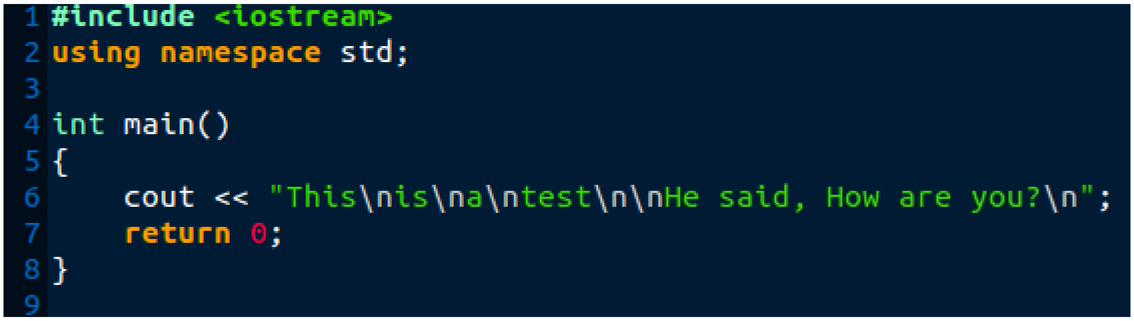
```
cout << "First sentence." << endl;
cout << "Second sentence." << endl;
```

Output

```
First sentence.
Second sentence.
```

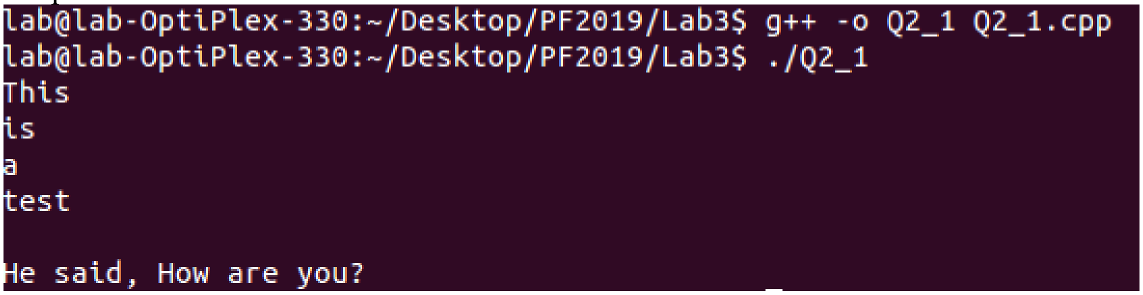
Example 2.1

Following program shows the use of Newline Escape Sequence (\n)



```
#include <iostream>
using namespace std;

int main()
{
    cout << "This\nis\na\ntest\n\nHe said, How are you?\n";
    return 0;
}

```

Output

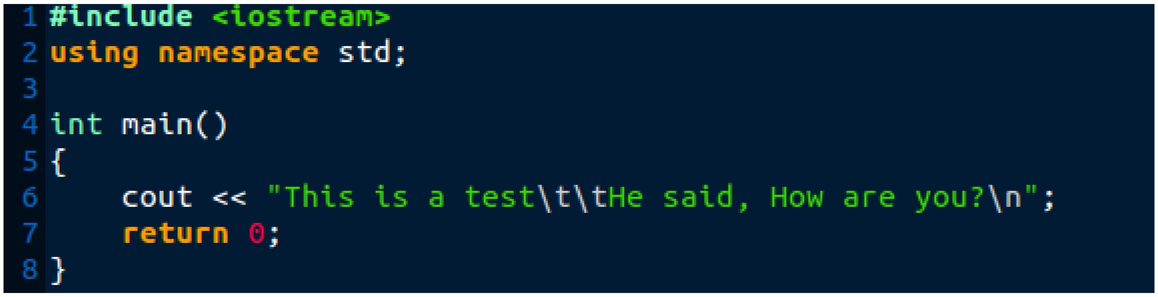


```
lab@lab-OptiPlex-330:~/Desktop/PF2019/Lab3$ g++ -o Q2_1 Q2_1.cpp
lab@lab-OptiPlex-330:~/Desktop/PF2019/Lab3$ ./Q2_1
This
is
a
test

He said, How are you?
```

Your turn: Edit above given code and use endl manipulator.

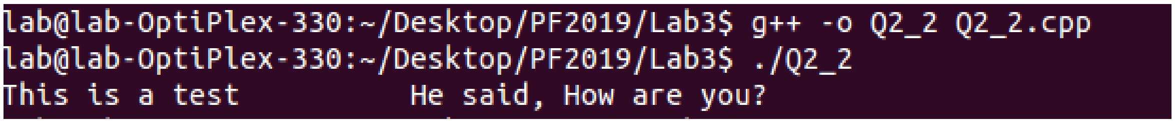
Example 2.2

This program shows the use of Horizontal tab Escape Sequence (\t)



```cpp
#include <iostream>
using namespace std;

int main()
{
    cout << "This is a test\t\tHe said, How are you?\n";
    return 0;
}
```

Output

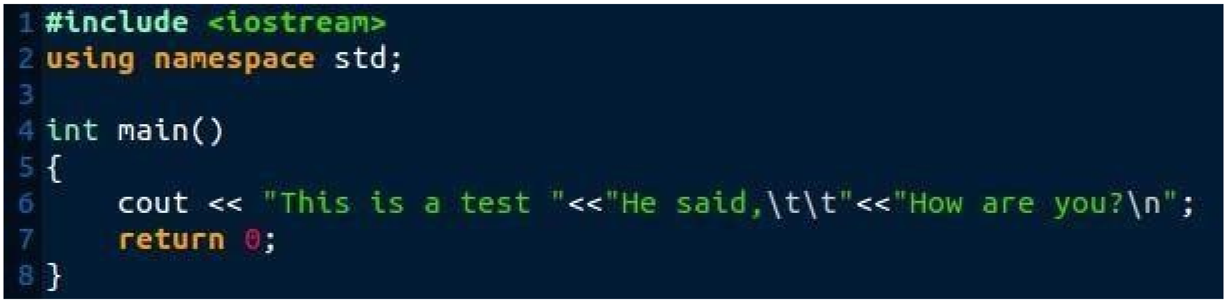


```
lab@lab-OptiPlex-330:~/Desktop/PF2019/Lab3$ g++ -o Q2_2 Q2_2.cpp
lab@lab-OptiPlex-330:~/Desktop/PF2019/Lab3$ ./Q2_2
This is a test          He said, How are you?
```

Now try escape sequences \\ , \' , \" yourself.

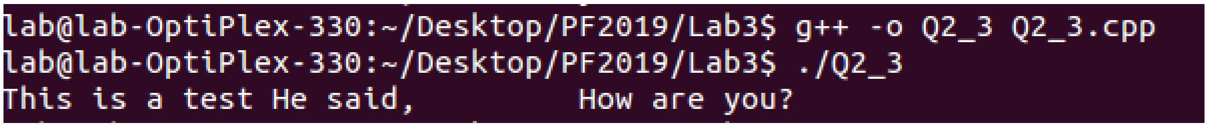
Example 2.3

Program using multiple insertion operations (<<)



```cpp
#include <iostream>
using namespace std;

int main()
{
    cout << "This is a test "<<"He said,\t\t"<<"How are you?\n";
    return 0;
}
```

Output



```
lab@lab-OptiPlex-330:~/Desktop/PF2019/Lab3$ g++ -o Q2_3 Q2_3.cpp
lab@lab-OptiPlex-330:~/Desktop/PF2019/Lab3$ ./Q2_3
This is a test He said,         How are you?
```

iomanip

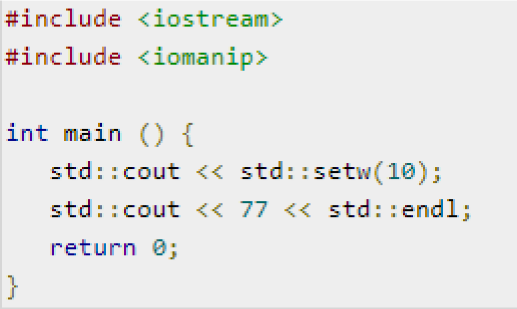
iomanip is a library that is used to manipulate the output of C++ program.

Below are some Parametric manipulators

1. *setw*

It is used to sets the field width to be used on output operations

<u>Example</u>



```
#include <iostream>
#include <iomanip>

int main () {
   std::cout << std::setw(10);
   std::cout << 77 << std::endl;
   return 0;
}
```

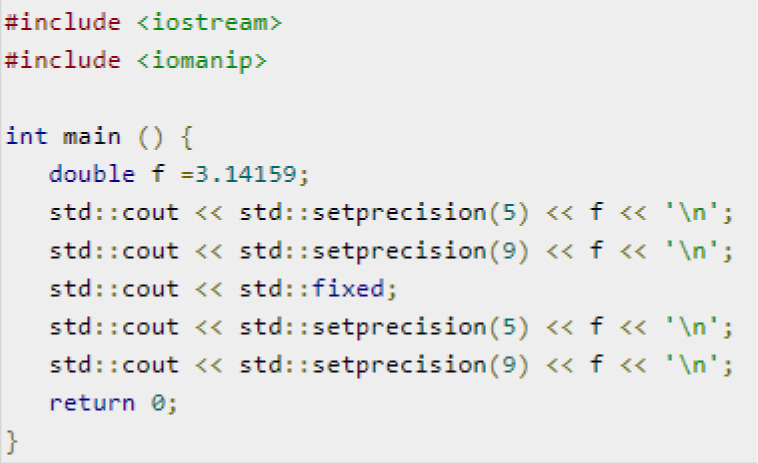
.

Now compile your code and see what the output is.

2. *setprecision*

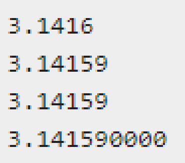
It is used to sets the decimal precision to be used to format floating-point values on output operations.

Example



```
#include <iostream>
#include <iomanip>

int main () {
   double f =3.14159;
   std::cout << std::setprecision(5) << f << '\n';
   std::cout << std::setprecision(9) << f << '\n';
   std::cout << std::fixed;
   std::cout << std::setprecision(5) << f << '\n';
   std::cout << std::setprecision(9) << f << '\n';
   return 0;
}
```

Output



```
3.1416
3.14159
3.14159
3.141590000
```

Tasks

Problem 01

Write a program to input your name as first name and last name, last 4 digits of roll number and CGPA and print that information.

Problem 02

Write a program that prints the following using COUT statement.

(1)

```
*

**
***

****
        ****
           ***
             **
               *
```

(2)

```
   *
  * *
 *   *
*     *
********
```

(3)

First Letter of
Your city using
Asterisk

(4)

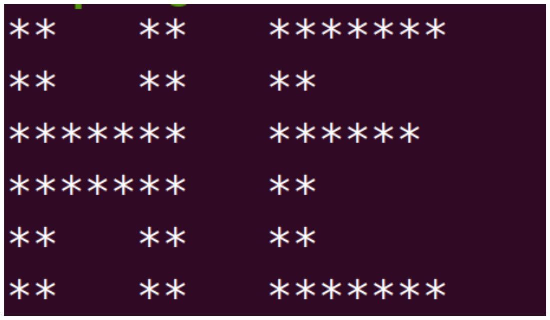
```
*         *
  *     *

     *

  *     *
 *        *
```

(5)

```
  *********
  *
*
  *
*
  *        *
  *********
```

Problem 03

Write a program that prints your second name (i.e. my second name is Toqeer) using COUT & setw statement.



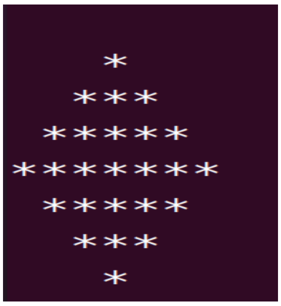
```
**     **    *******
**     **    **
*******      ******
*******      **
**     **    **
**     **    *******
```

Example

Note: Use **setw** function instead of space character

Problem 04

Write a program to print the following using just COUT statement & setw() function



```
   *
  ***
 *****
*******
 *****
  ***
   *
```

Note: Use **setw** function instead of space character

Submission Instructions:

1. Save all .cpp files with your task number

   e.g. Task01.cpp

2. Now create a new folder with name ROLLNO_LAB03 e.g. i22-XXXX-LAB03
3. Move all your .cpp files to this newly created directory and compress it into .zip file.
4. Now you must submit this zipped file on Google Classroom.
5. If you don’t follow the submission format marks will be deducted.