**Lab 04**

# 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++ lab@lab-OptiPlex-330:~$ **sudo apt install g++**

Note: g++ is already install on lab PCs

1. Create file of .cpp file extension using touch command ~$ **touch helloworld.cpp**

1. Now open the text editor using gedit command ~$ **gedit helloworld.cpp**

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



1. Save and close the file.
2. 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).



## 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 extracted data is stored. For example:

|  |
| --- |
| int age; cin >> age; |

1

2

**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); return 0;  } |

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

Enter a number: 9

Enter 2 numbers: 1 5

Sum = 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;

1. **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:



This produces the following output:



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

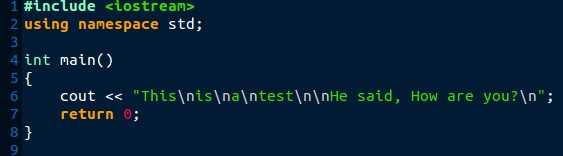


Output

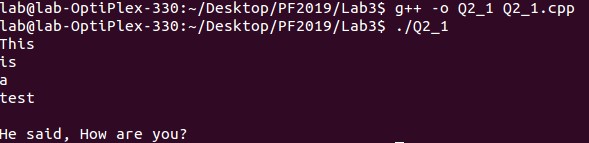


### Example 2.1

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



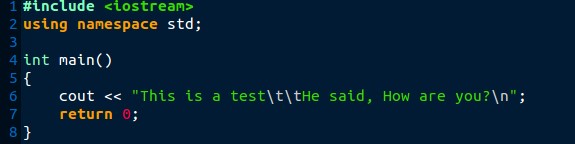
Output



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

### Example 2.2

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



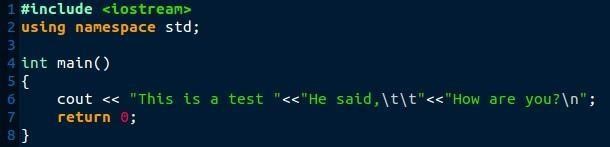
Output



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

### Example 2.3

Program using multiple insertion operations (<<)



## Output



**iomanip**

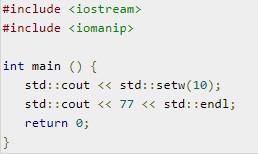
iomanip is a library that is used to manipulate the output of C++ program.

Below are some Parametric manipulators

### 1. setw

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

#### Example

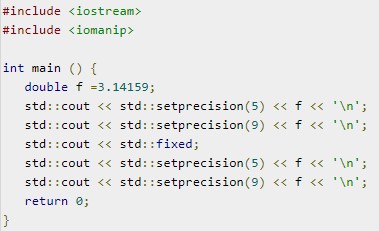
 .

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



### Output

c

**Exercise:**

## Problem 01

Write a program to input your name, roll number and CGPA and print that information.

## Problem 02

Write a program that prints the following using COUT statement & setw statement.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (1) | (2) | (3) | (4) | (5) |
| \* | \*\*\*\*\*\*\*\*\*\* | \* | \* | \* \* |
| \*\*\* | \* \* | \*\* | \*\* | \* \* |
| \*\*\*\*\* | \* \* | \*\*\* | \*\*\* | \* |
| \*\*\* | \* \* | \*\*\*\* | \*\*\*\* | \* \* |
| \* | \*\*\*\*\*\*\*\*\*\* | \*\*\*\*\* | \*\*\*\*\* | \* \* |

## Problem 03

Write a program that prints the following using COUT & setw statement.



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

|  |
| --- |
| **Submission Instructions:**   1. Save all .cpp files with your roll no and task number   e.g. i22XXXX\_Task01.cpp   1. Now create a new folder with name ROLLNO\_LAB04 e.g. i21XXXX\_LAB04 2. Move all your .cpp files to this newly created directory and compress it into .zip file. 3. Now you must submit this zipped file on Google Classroom. |