

# . Basic Input/Output in C++

C++ comes with libraries that provide us many ways for performing input and output. In C++ input and output is performed in the form of a sequence of bytes or more commonly known as **streams**.

**Input Stream:** If the direction of flow of bytes is from a device(for example Keyboard) to the main memory then this process is called input.

**Output Stream:** If the direction of flow of bytes is opposite, i.e. from main memory to device( display screen ) then this process is called output.

**Header files available in C++ for Input - Output operation are:**

- **iostream:** iostream stands for standard input output stream. This header file contains definitions to objects like cin, cout, cerr etc.
- **iomanip:** iomanip stands for input output manipulators. The methods declared in this files are used for manipulating streams. This file contains definitions of setw, setprecision etc.
- **fstream:** This header file mainly describes the file stream. This header file is used to handle the data being read from a file as input or data being written into the file as output.

In C++ articles, these two keywords **cout** and **cin** are used very often for taking inputs and printing outputs. These two are the most basic methods of taking input and output in C++. For using cin and cout we must include the header file *iostream* in our program.

In this article, we will mainly discuss the objects defined in the header file *iostream* like cin and cout.

- **Standard output stream (cout):** Usually the standard output device is the display screen. **cout** is the instance of the ostream class. cout is used to produce output on the standard output device which is usually the display screen. The data needed to be displayed on the screen is inserted in the standard output stream (cout) using the insertion operator (<<).

```
#include <iostream>
```

```
using namespace std;
```

```

int main( ) {

    char sample[] = "GeeksforGeeks";

    cout << sample << " - A computer science portal for geeks";

    return 0;

}

```

- **Output:**

```
GeeksforGeeks - A computer science portal for geeks
```

As you can see in the above program the insertion operator(<<) insert the value of the string variable **sample** followed by the string "A computer science portal for geeks" in the standard output stream **cout** which is then displayed on screen.

- **standard input stream (cin):** Usually the input device is the keyboard. cin is the instance of the class **istream** and is used to read input from the standard input device which is usually the keyboard. The extraction operator(>>) is used along with the object **cin** for reading inputs. The extraction operator extracts the data from the object **cin** which is entered using the keyboard.

```

#include<iostream>

using namespace std;

```

```

int main()

{

```

```

int age;

cout << "Enter your age:";

cin >> age;

cout << "\nYour age is: "<<age;

return 0;

}

```

- **Output:**

```

Enter your age:
Input : 18

Your age is: 18

```

The above program asks the user to input the age. The object cin is connected to the input device. The age entered by the user is extracted from cin using the extraction operator(>>) and the extracted data is then stored in the variable **age** present on the right side of the extraction operator.

- **Un-buffered standard error stream (cerr):** cerr is the standard error stream which is used to output the errors. This is also an instance of the ostream class. As cerr is un-buffered so it is used when we need to display the error message immediately. It does not have any buffer to store the error message and display later.

```
#include <iostream>
```

```
using namespace std;
```

```
int main( )

{

```

```
cerr << "An error occurred";
```

```
return 0;
```

```
}
```

- **Output:**

- An error occurred

- **buffered standard error stream (clog):** This is also an instance of the ostream class and used to display errors but unlike cerr the error is first inserted into a buffer and is stored in the buffer until it is not fully filled. The error message will be displayed on the screen too.

```
#include <iostream>
```

```
using namespace std;
```

```
int main( )
```

```
{
```

```
    clog << "An error occurred";
```

```
    return 0;
```

```
}
```

**Output:**

```
An error occurred
```

