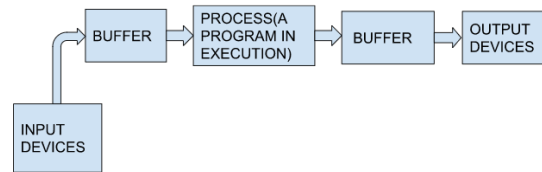
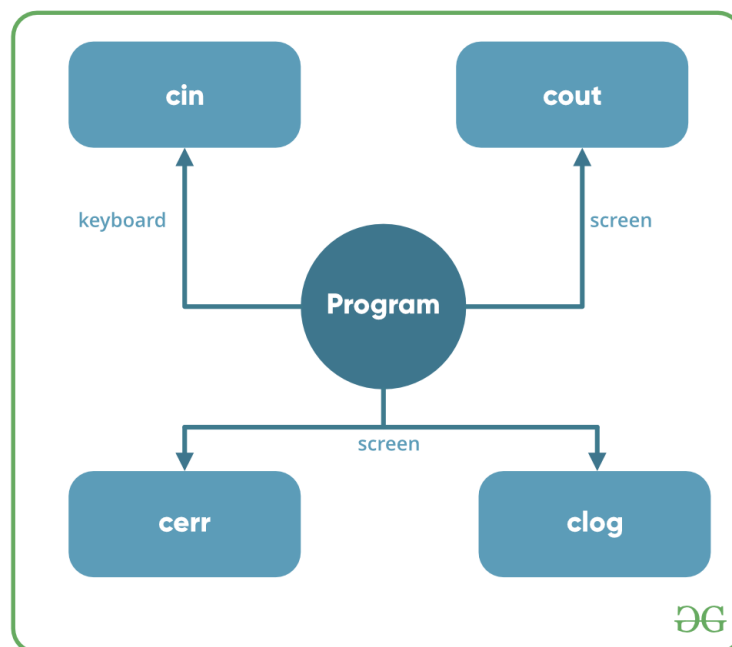


Input & Output in C++



C++ comes with libraries that provide us with many ways for performing input and output. In C++ input and output are 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 the 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 operations are:

1. **iostream:** iostream stands for standard input-output stream. This header file contains definitions of objects like cin, cout, cerr, etc.

2. **iomanip**: **iomanip** stands for input-output manipulators. The methods declared in these files are used for manipulating streams. This file contains definitions of `setw`, `setprecision`, etc.
3. **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.

The two instances **cout** in C++ and **cin** in C++ of `iostream` class are used very often for printing outputs and taking inputs respectively. These two are the most basic methods of taking input and printing output in C++. To use `cin` and `cout` in C++ one must include the header file `iostream` in the program.

- **Standard output stream (cout)**: Usually the standard output device is the display screen. The C++ **cout** statement is the instance of the `ostream` class. It 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
```

In the above program, the insertion operator(`<<`) inserts 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 the screen.

- **standard input stream (cin)**: Usually the input device in a computer is the keyboard. C++ `cin` statement is the instance of the class **istream** and is used to read input from the standard input device which is usually a 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;
}
```

Input :

18

Output:

```
Enter your age:
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):** The C++ cerr is the standard error stream that is used to output the errors. This is also an instance of the ostream class. As cerr in C++ is un-buffered so it is used when one needs to display the error message immediately. It does not have any buffer to store the error message and display it later.
- The main difference between cerr and cout comes when you would like to redirect output using “cout” that gets redirected to file if you use “cerr” the error doesn’t get stored in file.(This is what un-buffered means ..It cant store the message)

```
#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 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, or the buffer is not explicitly flushed (using flush()). 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