Manipulators: **Manipulators** are helping functions that can modify the input/output stream. It does not mean that we change the value of a variable, it only modifies the I/O stream using insertion (<<) and extraction (>>) operators. manipulators are simply an instruction to the output stream that modify the output in various ways

Advantages and Purpose of Manipulators

- It is mainly used to make up the program structure.
- Manipulators functions are special stream function that changes certain format and characteristics of the input and output.
- To carry out the operations of the manipulators <iomanip.h> must be included.
- Manipulators functions are specially designed to be used in conjunction with insertion (<<) and extraction (>>) operator on stream objects.
- Manipulators are used to changing the format of parameters on streams and to insert or extract certain special characters.

Types of Manipulators in C++

They are of two types one taking arguments and the other without argument.

1. Non-argument manipulators (Without Parameters)

Non-argument manipulators are also known as “Parameterized manipulators”. These manipulators require iomanip header. Examples are setprecision, setw and setfill.

2. Argumented manipulators (With parameters)

Argument manipulators are also known as “Non parameterized manipulators”. These manipulators require iostream header. Examples are endl, fixed, showpoint, left and flush.

Standard input/output Manipulators in C++

Here is the list of standard input/output **Manipulators** and their Functions in C++

- **setw (int n)** – To set field width to n
- **Setbase** – To set the base of the number system
- **stprecision (int p)** – The precision is fixed to p
- **setfill (Char f)** – To set the character to be filled
- **setiosflags (long l)** – Format flag is set to l
- **resetiosflags (long l)** – Removes the flags indicated by l
- **endl** – Gives a new line

- **skipws** – Omits white space in input
- **noskipws** – Does not omit white space in the input
- **ends** – Adds null character to close an output string
- **flush** – Flushes the buffer stream
- **lock** – Locks the file associated with the file handle
- **ws** – Omits the leading white spaces present before the first field
- **hex, oct, dec** – Displays the number in hexadecimal or octal or in decimal format

endl, setw, setfill,set precision.

endl is a function in Manipulators in C++:

The endl character introduces a new line or a line feed. It is analogous to the “n” character in the C computer language, and C++ supports the old line feed.

As an example,

```
cout << "This line use line feed as example" << endl;
```

```
 cout << number3 <<  endl << number4 << endl;
```

► setw () is a function in Manipulators in C++:

The setw() function is an output manipulator that inserts whitespace between two variables. You must enter an integer value equal to the needed space.

```
setw ( int n)
```

As an example,

```
cout << number5 << number6 << endl;
```

```
cout << setw(2) << number5 << setw(5) << number6 << endl;
```

Example Program:

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

int main()

{
  //variable declaration
int number1, number2, total;

//variable initialization
number1 = 100;
number2 = 345;
// expression

total = number1 + number2;
//printing output with setw

cout << endl;

cout << endl;
cout << setw(5) << number1 << " + " << setw(5) << number2 << " = "  << setw(6) <<
total << endl;
}
```

Output:

```
100 +   345 =    445
```

► setfill() is a function in Manipulators in C++:

It replaces setw(whitespaces)'s with a different character. It's similar to setw() in that it manipulates output, but the only parameter required is a single character. It's worth noting that a character is contained in single quotes.

```
setfill(char ch)
```

For example,

```
cout<< setfill('*') << endl;

cout << setw(5) << number5 << setw(5) << number6 << endl;
```

The output of the above will be ‘*’ character between variable *number5* and variable *number6*.

Example Program:

We will use the above setw() example with a little modifications.

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

using namespace std;
int main()

{
    //variable declaration

int number1, number2, total;

  //variable initialization
number1 = 100;
number2 = 345;

// expression

total = number1 + number2;

//printing output with setw
 cout << endl;
 cout << endl;
cout << setfill('*') << endl;
cout << setw(5) << number1 << " + " << setw(5) << number2 << " = "  << setw(6) <<
total << endl;

}
```

Output:

```
**100 + **345 = ***445
```

► setprecision() is a function in Manipulators in C++:

It is an output manipulator that controls the number of digits to display after the decimal for a floating point integer. Make careful to include the ipmanip header in your program because the function is defined there.

As an example,

```
float A = 1.34255;

cout << setprecision(3) << A << endl;
```

The output is 1.34.

Example Program:

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

using namespace std;
int main()
{

   //variable declaration
   float number1;
   //variable initialization

   number1 = 34.3358;
   //display the number using setprecision()

   cout << number1 << endl;
   cout << setprecision(2) << number1 << endl;
   cout << setprecision(3) << number1 << endl;
}
```