**IO Manipulation**

**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 special functions that can be included in the I/O statement to alter the format parameters of a stream.

Manipulators are operators that are used to format the data display.

To access manipulators, the file **iomanip.h** should be included in the program.

Some important manipulators in <ios> are:

* **showpos**: It forces to show a positive sign on positive numbers.
* **noshowpos**: It forces not to write a positive sign on positive numbers.

C++

// Example demonstrating the use of showpos and noshowpos

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

cout << showpos << 123 << endl; // Outputs: +123

cout << showpos << -456 << endl; // Outputs: -456

cout << noshowpos << 789 << endl; // Outputs: 789

}

* **showbase**: It indicates the numeric base of numeric values.

C++

// Code example demonstrating the use of showbase and noshowbase

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

cout << showbase << hex << 123 << endl; // Outputs: 0x7b

cout << showbase << oct << 456 << endl; // Outputs: 0o710

cout << showbase << dec << 789 << endl; // Outputs: 789

cout << noshowbase << hex << 345 << endl; // Outputs: 159

}

* **uppercase**: It forces uppercase letters for numeric values.
* **nouppercase:** It forces lowercase letters for numeric values.

C++

// Code example showing the use of uppercase and nouppercase

// lowercase and nolowercase would be having a similar behaviour

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

cout << uppercase << hex << 123 << endl; // Outputs: 0X7B

cout << nouppercase << hex << 456 << endl; // Outputs: 0xcd

cout << uppercase << oct << 789 << endl; // Outputs: 0O1145

cout << nouppercase << dec << 345 << endl; // Outputs: 345

}

* **fixed**: It uses decimal notation for floating-point values.
* **scientific**: It uses scientific floating-point notation.

C++

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

double x = 123.456789;

cout << fixed << x << endl; // Outputs: 123.456789

cout << scientific << x << endl; // Outputs: 1.234568e+02

}

* **hex**: Read and write hexadecimal values for integers and it works same as the setbase(16).
* **dec**: Read and write decimal values for integers i.e. setbase(10).
* **oct**: Read and write octal values for integers i.e. setbase(10).

C++

// Code implementation showing the use of hex, dec and oc

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

cout << hex << 123 << endl; // Outputs: 7b

cout << oct << 456 << endl; // Outputs: 710

cout << dec << 789 << endl; // Outputs: 789

}

* **left**: It adjusts output to the left.
* **right**: It adjusts output to the right.
* **setfill:** The setfill manipulator is used in C++ to set the fill character that is used to pad a field when outputting a value using the setw manipulator. The setfill manipulator is typically used in conjunction with the setw manipulator, which sets the field width for output.
* **setw:** In C++, the setw manipulator is used to set the width of the output field for certain types of data. When outputting data to the console or to a file, you can use setw to specify the minimum number of characters that should be used to represent the output data. This can be useful for formatting the output and aligning columns of data.

C++

// Code implementation showing the use of left, right, setfill, setw

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

cout << left << setw(10) << setfill('-') << "Apple" << endl; // Output: Apple-----

cout << right << setw(10) << setfill('-') << "Orange" << endl; // Output: ----Orange

}

The ***left*** manipulator causes the output to be left-aligned within the specified width. The ***right*** manipulator causes the output to be right-aligned within the specified width.

The ***setw*** manipulator sets the width of the output field, and the ***setfill*** manipulator sets the character that should be used to fill any unused space in the output field.

In the example above, the ***setw*** manipulator was used to set the width of the output field to 10 characters, and the ***setfill*** manipulator was used to set the fill character to a dash. The ***left*** manipulator caused the word "Apple" to be left-aligned within the field, and the ***right*** manipulator caused the word "Orange" to be right-aligned within the field. This caused the output to be aligned in two neat columns.