

## C++ 5\_1

### 5.1 cout (output stream)

On most program environments, the standard output by default is the screen, and the C++ stream object defined to access it is cout. cout is an instance of ostream class

For formatted output operations, cout is used together with the insertion operator, which is written as << (i.e., two "less than" signs).

```
cout << "this is Output";  
// prints this is Output sentence on screen  
cout << 50;  
// prints number 50 on screen  
cout << x;  
// prints the value of x on screen
```

The << operator inserts the data that follows it into the stream that precedes it. In the examples above, it inserted the literal string Output sentence, the number 120, and the value of variable x into the standard output stream cout. Notice that the sentence in the first statement is enclosed in double quotes (") because it is a string literal, while in the last one, x is not. The double quoting is what makes the difference; when the text is enclosed between them, the text is printed literally; when they are not, the text is interpreted as the identifier of a variable, and its value is printed instead.

For example, these two sentences have very different results:

```
cout << "Hello"; // prints Hello  
cout << Hello; // prints the content of variable Hello
```

Multiple insertion operations (<<) may be chained in a single statement:

```
cout << "This " << " is a " << "single C++ statement";
```

This last statement would print the text This is a single C++ statement. Chaining insertions is especially useful to mix literals and variables in a single statement:

```
cout << "I am " << age << " years old and my zipcode is " << zipcode;
```