```
/*
  * C++ Programming Notes
  * Ravi Kumar Reddy K
  * github.com/ravikumark815
*/
```

## Preset:

- Invented by Bjarne Stroustrup in 1979
- Middle Level Language
- Versions: C++ 14, C++11, C++99

## Hello World:

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

int imGlobal = 0;
const double PI = 3.141;

int main(int argc, char**argv) {
  cout << "Hello World\n";
  return 0;
}
```

- Namespaces
- main: Start executing from here
- Cout allows us to output information to console
- “<<” Stream insertion operator: Takes string on the right to cout stream
- “endl” Issue newline and force write to console
- argc: No of arguments passed to main
- argv: Array of pointers to strings in the arg vector
- int: Return an integer when done executing
- imGlobal: Global variable and accessible everywhere else.
- const double PI: Global variable whose value cannot be changed anywhere else

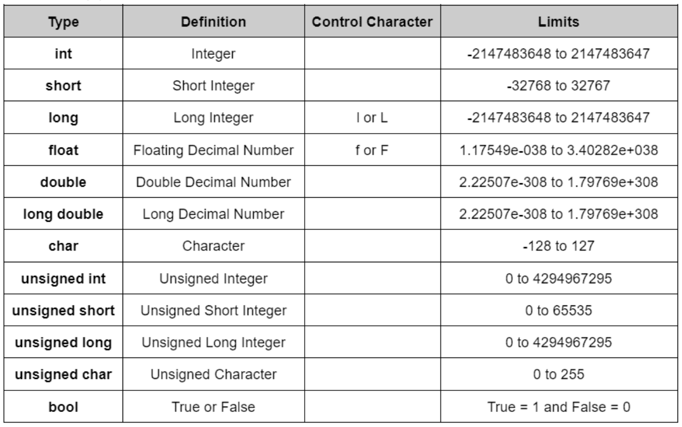
## Comments:

```
/*
Multi
Line
Comment
*/
// Single Line Comment
```

## Common Header files:

- #include <cstdlib> // Sorting, Searching, import c libraries, rand, memmgmt, and other general-purpose functions
- #include <iostream> // Read and Write data
- #include <string> // Work with strings
- #include <limits> // Min and max values
- #include <vector> // Work with vectors
- #include <sstream> // Work with string streams
- #include <numeric> // Work with sequences of values
- #include <ctime> // Work with time
- #include <cmath> //Common math functions

## Data Types:

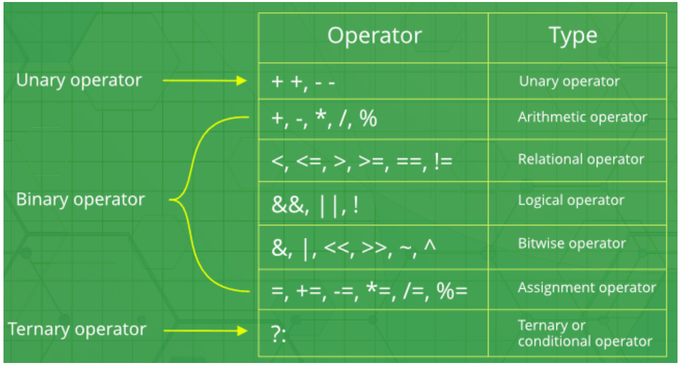


| Type | Definition | Control Character | Limits |
|---|---|---|---|
| int | Integer | | -2147483648 to 2147483647 |
| short | Short Integer | | -32768 to 32767 |
| long | Long Integer | l or L | -2147483648 to 2147483647 |
| float | Floating Decimal Number | f or F | 1.17549e-038 to 3.40282e+038 |
| double | Double Decimal Number | | 2.22507e-308 to 1.79769e+308 |
| long double | Long Decimal Number | | 2.22507e-308 to 1.79769e+308 |
| char | Character | | -128 to 127 |
| unsigned int | Unsigned Integer | | 0 to 4294967295 |
| unsigned short | Unsigned Short Integer | | 0 to 65535 |
| unsigned long | Unsigned Long Integer | | 0 to 4294967295 |
| unsigned char | Unsigned Character | | 0 to 255 |
| bool | True or False | | True = 1 and False = 0 |

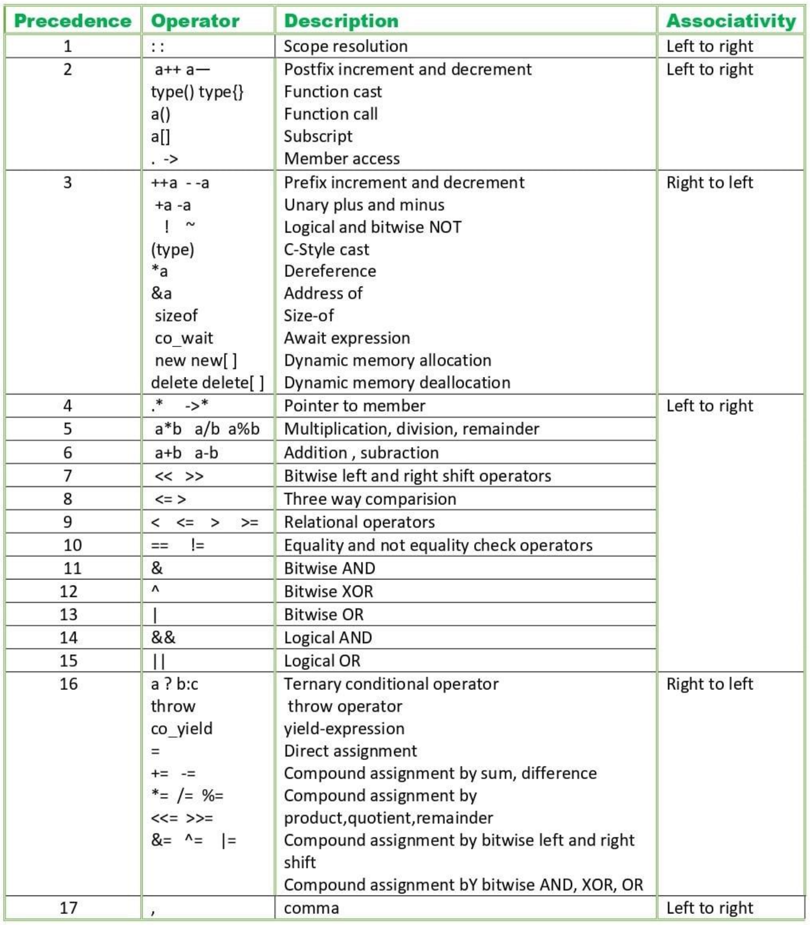
## Input and Output:

- cout << “Min int” << numeric_limits<int>::min();
- cout << “Max short int” << numeric_limits<short int>::max();
- printf(“Sum = %.7f\n”), (1.1111111+1.1111111)); // To print formatted output of float upto 7 decimal places
- cout << “int Byte:” << sizeof(int) << endl;
- printf(“%c %d %5d %.3f %s\n”, ‘A’, 10, 5, 3.1234, “Hi); // O/p: A 10      5 3.123 Hi //Right justify
- cin >> num_str; //to take in input for num1
- int num1 = stoi(num_str) //To convert num1 from string to int;
- bool res=true; cout.setf(ios::boolalpha); cout << res << endl; // To print booleans

## Operators, Precedence, Associativity:

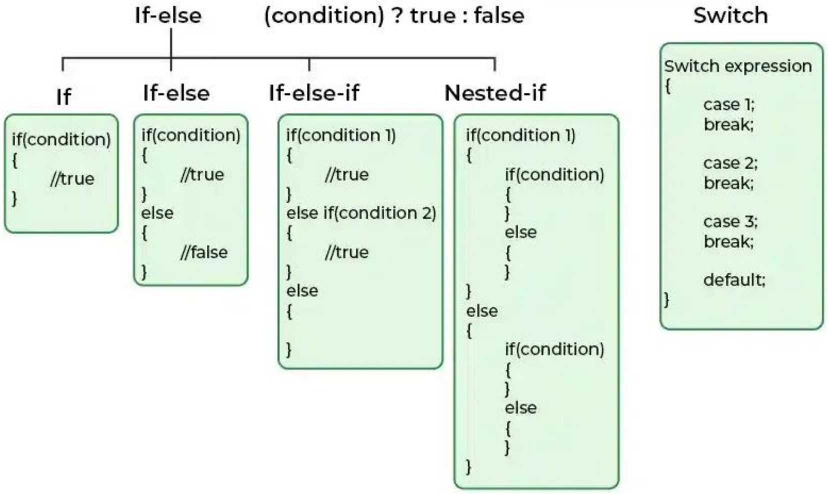

Unary operator
Binary operator
Ternary operator

| Operator | Type |
|---|---|
| + +, - - | Unary operator |
| +, -, *, /, % | Arithmetic operator |
| <, <=, >, >=, ==, != | Relational operator |
| &&, \| \|, ! | Logical operator |
| &, \|, <<, >>, ~, ^ | Bitwise operator |
| =, +=, -=, *=, /=, %= | Assignment operator |
| ?: | Ternary or conditional operator |



| Precedence | Operator | Description | Associativity |
|---|---|---|---|
| 1 | : : | Scope resolution | Left to right |
| 2 | a++ a—<br>type() type{}<br>a()<br>a[]<br>. -> | Postfix increment and decrement<br>Function cast<br>Function call<br>Subscript<br>Member access | Left to right |
| 3 | ++a - -a<br>+a -a<br>! ~<br>(type)<br>*a<br>&a<br>sizeof<br>co_wait<br>new new[ ]<br>delete delete[ ] | Prefix increment and decrement<br>Unary plus and minus<br>Logical and bitwise NOT<br>C-Style cast<br>Dereference<br>Address of<br>Size-of<br>Await expression<br>Dynamic memory allocation<br>Dynamic memory deallocation | Right to left |
| 4 | .* ->* | Pointer to member | Left to right |
| 5 | a*b a/b a%b | Multiplication, division, remainder | |
| 6 | a+b a-b | Addition , subraction | |
| 7 | << >> | Bitwise left and right shift operators | |
| 8 | <= > | Three way comparision | |
| 9 | < <= > >= | Relational operators | |
| 10 | == != | Equality and not equality check operators | |
| 11 | & | Bitwise AND | |
| 12 | ^ | Bitwise XOR | |
| 13 | \| | Bitwise OR | |
| 14 | && | Logical AND | |
| 15 | \|\| | Logical OR | |
| 16 | a ? b:c<br>throw<br>co_yield<br>=<br>+= -=<br>*= /= %=<br><<= >>=<br>&= ^= \|= | Ternary conditional operator<br>throw operator<br>yield-expression<br>Direct assignment<br>Compound assignment by sum, difference<br>Compound assignment by product,quotient,remainder<br>Compound assignment by bitwise left and right shift<br>Compound assignment bY bitwise AND, XOR, OR | Right to left |
| 17 | , | comma | Left to right |

**Conditional Statements:**



```
if(condition)
{
      //true
}
```

```
if(condition)
{
      //true
}
else
{
      //false
}
```

```
if(condition 1)
{
      //true
}
else if(condition 2)
{
      //true
}
else
{

}
```

```
if(condition 1)
{
      if(condition)
      {
      }
      else
      {
      }
}
else
{
      if(condition)
      {
      }
      else
      {
      }
}
```

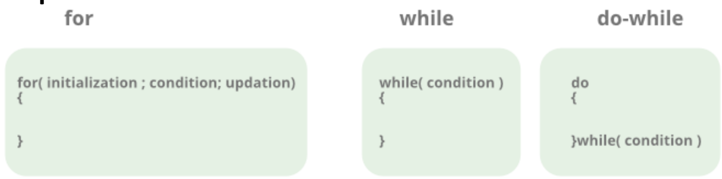
```
Switch expression
{
      case 1;
      break;

      case 2;
      break;

      case 3;
      break;

      default;
}
```

**Loops:**



```
for( initialization ; condition; updation)
{

}
```

```
while( condition )
{

}
```

```
do
{

}while( condition )
```

**Arrays:**

```
void main(int argc, char**argv) {
        int array1 [10] = {1};      // Size
```

```
    int array2 [] = {1,2,3};     // Size for this would automatically be 3
    int array3 [5] = {8,9};      //
    cout << “First val: ” << array1[0] << endl;
    array1[0] = 7;
    int array4[2][3][3] = { {{1,2}, {3,4}}, {{5,6}, {7,8}} };    // Multidimensional arrays
    cout << array4[0][1][1] <<endl //prints 4
    return 0;
}
```

- Size once defined cannot be changed.

**Vectors:**

- Used when size of arrays cannot be determined in precursor.

```
void main() {
    vector<int> vnums(2);
    vnums[0] = 1;
    vnums[1] = 2;
    vnums.push_back(3);
    cout << “Size:” << vnums.size() << endl; //O/P Size: 3
}
```