Techimax

Fundamentals of Programming in C++

Operators

```
//unary
int a = 5;
int b = -3;
int a1 = -a;
int a2 = +b;
```

```
int x = 4;
++x; // Pre-Increment
x++; // Post-Increment
--x; // Pre-Decrement
x--; // Post-Decrement
```

```
//binary arithmatic
int c1 = a + b;
int c2 = a - b;
int c3 = a * b;
int c4 = a / b;
float c5 = (float)a / b;
int c6 = a % b;

int d1 = a;
d1 += a;
d1 -= a;
d1 /= a;
d1 *= a;
d1 %= a;
```

Operator Precedence
https://en.cppreference.com/w/cpp/language/operator_precedence

Understanding Increment-Decrement

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

int main()
{
    int x = 5, y = 5;
    cout << x << " " << y << '\n';
    cout << ++x << " " << --y << '\n'; // Prefix
    cout << x << " " << y << '\n';
    cout << x++ << " " << y-- << '\n'; // Postfix
    cout << x << " " << y << '\n';
}
```

```
//output
5 5
6 4
6 4
6 4
7 3
```

Relational Operators

```
//relational operators
      x < y
      x > y
      x >= y
      x <= y
      x == y
      x != y
```

Operator	Symbol	Form	Operation
Greater than	>	x > y	true if x is greater than y, false otherwise
Less than	<	x < y	true if x is less than y, false otherwise
Greater than or equals	>=	x >= y	true if x is greater than or equal to y, false otherwise
Less than or equals	<=	x <= y	true if x is less than or equal to y, false otherwise
Equality	==	x == y	true if x equals y, false otherwise
Inequality	!=	x != y	true if x does not equal y, false otherwise

Logical Operators

Operator	Symbol	Form	Operation
Logical NOT	!	!x	true if x is false, or false if x is true
Logical AND	&&	x && y	true if both x and y are true, false otherwise
Logical OR	||	x || y	true if either x or y are true, false otherwise

Bitwise Operators

Operator	Symbol	Form	Operation
left shift	<<	x << y	all bits in x shifted left y bits
right shift	>>	x >> y	all bits in x shifted right y bits
bitwise NOT	~	~x	all bits in x flipped
bitwise AND	&	x & y	each bit in x AND each bit in y
bitwise OR	|	x | y	each bit in x OR each bit in y
bitwise XOR	^	x ^ y	each bit in x XOR each bit in y

Operator	Symbol	Form	Operation
Left shift assignment	<<=	x <<= y	Shift x left by y bits
Right shift assignment	>>=	x >>= y	Shift x right by y bits
Bitwise OR assignment	|=	x |= y	Assign x | y to x
Bitwise AND assignment	&=	x &= y	Assign x & y to x
Bitwise XOR assignment	^=	x ^= y	Assign x ^ y to x

Conditionals (if-elseif-else)

```
if (/* condition == True */)
{
  // do This
}
```

```
if (/* condition == True */)
{
  // do This
}
else // if condition == False
{
  // do This
}
```

```
if (/* condition == True */)
{
  // do This
}
else if (/* conditionB == True */)
{
    // do This
}
else // if condition == False
{
    // do This
}
```

Conditionals (switch)

```
char grade = 'D';
switch (grade)
{
case 'A':
    cout << "Excellent!\n";
    break;
case 'B':
    cout << "Awesome!\n";
    break;
case 'C':
    cout << "Well done";
    break;
case 'D':
    cout << "You passed";
    break;
case 'F':
    cout << "Better try again";
    break;
default:
    cout << "Invalid grade";
}
```

Functions

```
return_type function_name( parameter list )
{
   body of the function
}
```

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

// No parameters
// No Return Type (ie. void)
void sayHey()
{
    cout << "Hey There !\n";
}

int main()
{
    sayHey();
}
```

Functions

```
return_type function_name( parameter list )
{
   body of the function
}
```

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

// Two parameters of int type
// No Return Type (ie. void)
void printSum(int a, int b)
{
    cout << a << " + " << b << " = " << a + b << '\n';
}

int main()
{
    printSum(-7, 4);
}
```

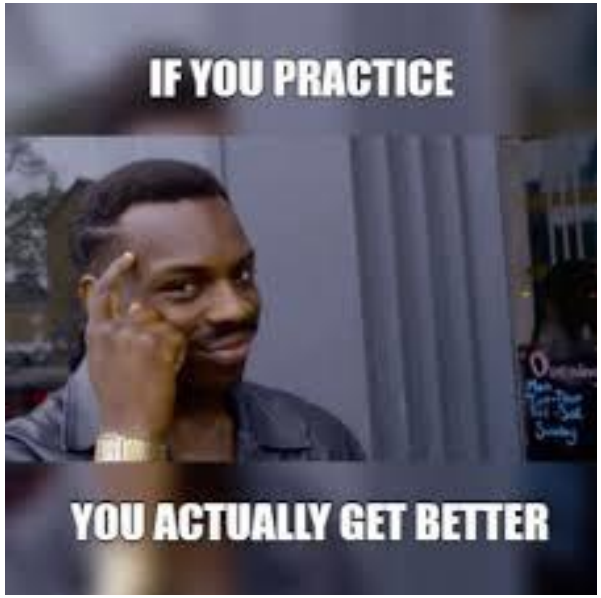
Functions

```
return_type function_name( parameter list )
{
   body of the function
}
```

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

// Two parameters of int type
// int Return Type
int maxOfTwo(int a, int b)
{
    if (a > b)
        return a;
    else
        return b;
}

int main()
{
    int maxNum = maxOfTwo(3, 5);
    cout << maxNum << '\n';
}
```



Given the coefficients of the quadratic equation

$$ax^2 + bx + c = 0$$

find its roots using the quadratic formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```cpp
#include <iostream>
#include <math.h>
using namespace std;

int main()
{
    double a, b, c, D, x1, x2;
    cout << "Enter the coefficients a, b, & c : ";
    cin >> a >> b >> c;
    cout << "\n" << a << "x^2 + " << b << "x + " << c << " = 0" << "\n\n";

    if (a == 0)
    {
        cout << "The given equation is not a quadratic equation :( \n\n";
        return;
    }

    D = b * b - 4 * a * c;
    if (D == 0)
    {
        cout << "Determinant is " << D << ". \tRoots are real and equal\n";
        x1 = -b / (2 * a);
        // x2 = x1;
        cout << "x1 = x2 = " << x1 << endl;
    }
    else if (D > 0)
    {
        cout << "Determinant is " << D << ". \tRoots are real and different\n";
        x1 = (-b + sqrt(D)) / (2 * a);
        x2 = (-b - sqrt(D)) / (2 * a);
        cout << "x1 = " << x1 << "\nx2 = " << x2 << endl;
    }
    else //if D < 0
    {
        cout << "Determinant is " << D << ". \tRoots are imaginary\n";
        double real_part = -b / (2 * a);
        double img_part = sqrt(abs(D)) / (2 * a);
        cout << "x1 = " << real_part << " + " << img_part << " i " << endl;
        cout << "x2 = " << real_part << " - " << img_part << " i " << endl;
    }
}
```