

Programming and Problem solving

Lecture 4

Israa S. Al-Furati

Programming in C++

There are multiple compilers and text editors could be used to run C++ programming. These may differ from system to system. We will use CodeBlockes editor in this course.

3.1 Basic Input/Output

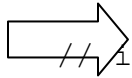
Cin : standard input stream

```
1 int age;  
2 cin >> age
```

Cout : standard output stream

```
1 cout << "Output sentence"; // prints Output sentence on  
screen  
2 cout << 120;                // prints number 120 on screen  
3 cout << x;                  // prints the value of x on  
screen
```

Simple input/output program:



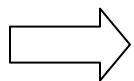
//input output example

```
#include <iostream>  
using namespace std;  
  
int main ()  
{  
    int a,b;  
    cout << "Please enter the first number: ";  
    cin >> a;  
    cout << " Please enter the second number: "<< b;  
    cin>> b;  
  
    return 0;  
}
```

3.2 Arithmetic Operators

There are following arithmetic operators supported by C++ language –
Assume variable A holds 10 and variable B holds 20, then

| Operator | Description | Example |
|----------|---|---------------------|
| + | Adds two operands | A + B will give 30 |
| - | Subtracts second operand from the first | A - B will give -10 |
| * | Multiplies both operands | A * B will give 200 |
| / | Divides numerator by denominator | B / A will give 2 |
| % | Modulus Operator and remainder of after an integer division | B % A will give 0 |
| ++ | Increment operator, increases integer value by one | A++ will give 11 |
| -- | Decrement operator, decreases integer value by one | A-- will give 9 |



```
// C++ arithmetic
#include <iostream>
using namespace std;
int main()
{
    float biscuit;
    int babies;

    cout << "Enter a number: ";
    cin >> biscuit;
    cout << "Enter another number: ";
    cin >> babies;

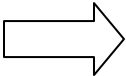
    cout << " biscuit = " << biscuit << "; babies = " << babies << endl;
    cout << " biscuit + babies = " << biscuit + babies << endl;
    cout << " biscuit - babies = " << biscuit - babies << endl;
    cout << " biscuit * babies = " << biscuit * babies << endl;
    cout << " biscuit / babies = " << biscuit / babies << endl;
```

```
return 0;}
```

3.3 Relational Operators

There are following relational operators supported by C++ language
Assume variable A holds 10 and variable B holds 20, then –

| Operator | Description | Example |
|----------|---|-----------------------|
| == | Checks if the values of two operands are equal or not, if yes then condition becomes true. | (A == B) is not true. |
| != | Checks if the values of two operands are equal or not, if values are not equal then condition becomes true. | (A != B) is true. |
| > | Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true. | (A > B) is not true. |
| < | Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true. | (A < B) is true. |
| >= | Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true. | (A >= B) is not true. |
| <= | Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true. | (A <= B) is true. |



```
// relational operators  
#include<iostream>
```

```
using namespace std;

int main()
{
int a=10,b=20,c=10;

if(a>b)
    cout<<"a is greater"<<endl;

if(a<b)
    cout<<"a is smaller"<<endl;

if(a<=c)
    cout<<"a is less than/equal to c"<<endl;

if(a>=c)
    cout<<"a is less than/equal to c"<<endl;

return 0;}
```

3.4 Logical Operators

There are following logical operators supported by C++ language.
Assume variable A holds 1 and variable B holds 0, then –

| Operator | Description | Example |
|----------|---|--------------------|
| && | Called Logical AND operator. If both the operands are non-zero, then condition becomes true. | (A && B) is false. |
| | Called Logical OR Operator. If any of the two operands is non-zero, then condition becomes true. | (A B) is true. |
| ! | Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true, then Logical NOT operator will make false. | !(A && B) is true. |

// Logical Operators
#include <iostream>
using namespace std;

```
int main()
{
    cout << "Enter a number: ";
    int value;
    cin >> value ;

    if (value > 10 && value < 20)
        cout << "Your value is between 10 and 20" << endl;
    else
        cout << "Your value is not between 10 and 20" << endl;
    return 0;
}
```

3.5 Bitwise Operators

Bitwise operator works on bits and perform bit-by-bit operation. The truth tables for &, |, and ^ are as follows –

| p | q | p & q | p q | p ^ q |
|---|---|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |

Assume if A = 60; and B = 13; now in binary format they will be as follows

A = 0011 1100

B = 0000 1101

A&B = 0000 1100

A|B = 0011 1101

A^B = 0011 0001

~A = 1100 0011

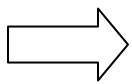
→ // (~) one's compliment operator
#include <iostream>

```
using namespace std;

int main()
{
    // 12 = 0000 1100
    unsigned int num1 = 12
    int num2 = 0;

    num2 = ~num1;
    cout << "Value of num2 is: " << num2 << endl ;

    return 0;
}
```



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

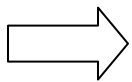
int main()
{
    unsigned int num1 = 10;           // 10 = 0000 1010
    unsigned int num2 = 12;           // 12 = 0000 1100
    int num3 = 0;

    num3 = num1 & num2;               // 8 = 0000 1000
    cout << "Value of num3 is : " << num3 << endl ;

    return 0;
}
```

3.6 Mathematical Functions

C++ provides various mathematical functions like `log()`, `modf()`, `pow()`, `sqrt()`, `sin()`, `cos()`, `abs()` etc. that aid in mathematical calculations. `<math.h>` library should be called.



Example shows few of the mathematical operations:

```
// C++ Mathematical Functions
#include<iostream.h>
#include<conio.h>
#include<math.h>
int main()
{
```



```
Short int si = 100;
int i = -1000;
long int li = 8;
float f = 230.47;
double d = 200.347;

cout<<"sqrt(si): "<<sqrt(si)<<endl;
cout<<"pow(li, 3): "<<pow(li, 3)<<endl;
cout<<"sin(d): "<<sin(d)<<endl;
cout<<"abs(i) : "<<abs(i)<<endl;
cout<<"floor(d): "<<floor(d)<<endl;
cout<<"sqrt(f): "<<sqrt(f)<<endl;
cout<<"pow(d, 2): "<<pow(d, 2)<<endl;

    return 0;
}
```

The output of the above C++ program

```
sqrt(si): 10
pow(li, 3): 512
sin(d): -0.6555
abs(i) : 1000
floor(d): 200
sqrt(f): 15.181
pow(d, 2): 40138.92
```