

Lecture 8
Introduction to C++
by Hafijur Rahman

C vs C++

C++ was originally developed (inventor: Bjarne Stroustrup (Bell Laboratories)) to be the next version of C, not a new language. C++ started with the name “***C with Classes***”.

Similarities (C & C++)

Same built-in data types

Same compiler preprocessor

handles #include & #define

Same built-in operators on primitive types (+-/*.....)

Same built-in control structures

if, for, while, switch

Must have a function names “main” to determine where the program starts

Functions are defined the same way

Programs can be split up into separate files

C vs C++

Differences (C & C++)

In C, all code exists in function and the paradigm is that you manipulate data with functions.

In C++, classes are used to model the behavior of data objects, behavior is represented by functions and those functions are used to modify the object's data

In C, I/O is accomplished via library functions

In C++, I/O is done by using object methods.

C++ has function overloading (two functions may have the same name)

In C, function names must be unique.

C++ has operator overloading (operators can be redefined to do other things)

C vs C++

Simple C Program	Simple C++ Program
<pre># include <stdio.h> int main(void) { printf("Hello World\n"); return 0; }</pre>	<pre>#include <iostream> using namespace std; int main(void) { cout << "Hello World!\n"; return 0; }</pre>

C++ is a superset of C, supporting almost everything about C. C++ provides ***object-oriented programming (OOP)***. A C++ compiler will always successfully compile the C source code, the reverse is not true.

Input statements

cin >> variable-name;

Meaning: read the value of the variable called <variable-name> from the user

Example:

```
cin >> a;
```

```
cin >> b >> c;
```

```
cin >> x;
```

```
cin >> my-character;
```

Output statements

cout << variable-name;

Meaning: print the value of variable <variable-name> to the user

cout << “any message “;

Meaning: print the message within quotes to the user

cout << endl;

Meaning: print a new line

Example:

```
cout << a;
```

```
cout << b << c;
```

```
cout << “This is my character: “ << my-character << “ he he he” << endl;
```

Example 2 (Procedural Approach)

```
// Simple stream input/output
#include <iostream>
using namespace std;
int main()
{
    cout << "Enter your age: ";
    int myAge;
    cin >> myAge;
    cout << "Enter your friend's age: ";
    int friendsAge;
    cin >> friendsAge;
```

```
if (myAge > friendsAge)
    cout << "You are older.\n";
else
    if (myAge < friendsAge)
        cout << "You are younger.\n";
    else
        cout << "You and your friend are the
same age.\n";

return 0;
}
```


Example 1 (Procedural Approach)

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

int main() {
    int num1 = 8;
    int num2 = 88;
    int sum1, sum2;
    sum1 = num1 + num2;
    cout << sum1 << endl;

    sum2 = num1 + num2;
    cout << sum2 << endl;

    int count = 88;
    int sum3 = 0;
    for (int i = 0; i < count; i++) {
        sum3 += i;
    }
}
```

```
cout << sum3 << endl;
```

```
count = 99;
```

```
int sum4 = 0;
```

```
for (int i = 0; i < count; i++) {
```

```
    sum4 += i;
```

```
}
```

```
cout << sum4 << endl;
```

```
}
```

THANKS