



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

200
1822 - 2022

Review of C++ Programming

Man Geni Addysg Uwch yng Nghymru
Where Higher Education Began in Wales

www.uwtsd.ac.uk

Contents

- Gentle introduction to C++
- File I/O
- Control statements
- Arrays

About learning C++

C++ is a complicated computer language

It takes time to master C++ – one year is the rule of thumb

C++ exposes you to lots of “low-level details” – these are hidden in languages like Java, Matlab and Python

Hopefully, you will appreciate the speed and flexibility of C++

Recommended attitude

- Dive into executable examples
- Don't try to understand everything
- Try to adapt examples to new problems
- Look up technical details in manuals/textbooks

Why do you need to learn “old” compiled languages?

- Because C, C++, and Fortran (77/95) are the most efficient existing tools for intensive numerical computing
- Because tons of fast and well-tested codes are available in Fortran, C/C++ (game development)
- Newer languages have emphasized simplicity and reliability – at the cost of computational efficiency
- To get speed, you need to dive into the details of compiled languages, and this course is a first, gentle step

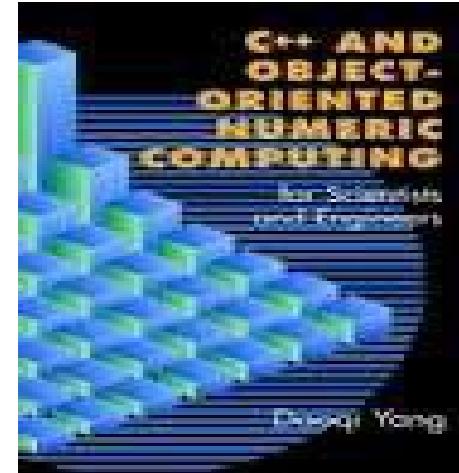
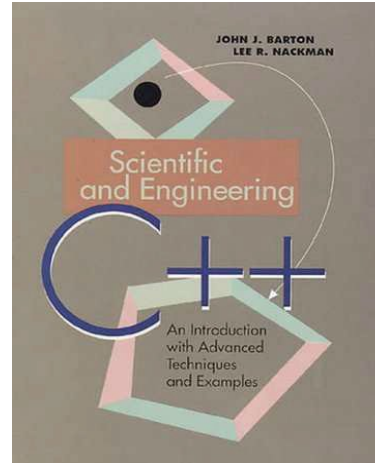
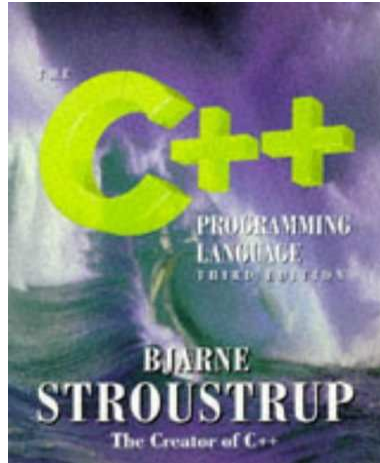
C++

C++ extends C with

- nicer syntax
- declare variables wherever you want
- in/out function arguments use references (instead of pointers)
- classes for implementing user-defined data types
- a standard library (STL) for frequently used data types (list, stack, queue, vector, hash, string, complex, ...)
- object-oriented programming
- exceptions for error handling

Recommended C++ textbooks

- Stroustrup, Barton & Nackman, or Yang:



More books reviewed:

<http://www.accu.org/>

<http://www.comeaucomputing.com/booklist/>



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

200
1822 - 2022

Man Geni Addysg Uwch yng Nghymru
Where Higher Education Began in Wales

Output Text pattern

Goal: make a window on the screen with the text “Hello World”

Goal: make a window on the screen with the text "Hello World"

```
/*  
 * Multiple line comment  
 * classic program to print hello world  
 */  
#include<iostream>  
//Single line comment  
using namespace std;  
//This is where the execution of program begins  
int main()  
{  
    // displays Hello World! on screen  
    cout<<"Hello World!";  
    return 0; }
```

Goal: make a window on the screen with the text "Hello World"

```
/*  
 * Multiple line comment  
 * classic program to print hello world  
 */  
  
#include<iostream>  
//Single line comment  
using namespace std;  
  
//This is where the execution of program begins  
int main()  
{  
    // displays Hello World! on screen  
    cout<<"Hello World!";  
    return 0;  
}  
//End of program
```

```
std::cout << ""hello World"  
  
// read from keyboard into r  
std::cin >> r;
```

Goal: make a window on the screen with the text "Hello World"

```
/*  
 * Multiple line comment  
 * classic program to print hello world  
 */  
  
#include<iostream>  
  
//Single line comment  
using namespace std;  
  
//This is where the execution of program begins  
int main()  
{  
    // displays Hello World! on screen  
    cout<<"Hello World!";  
    return 0; }  

```

Goal: make a window on the screen with the text "Hello World"

```
/*  
 * Multiple line comment  
 * classic program to print hello world  
 */  
  
#include<iostream>  
  
//Single line comment  
using namespace std;  
  
//This is where the execution of program begins  
int main()  
{  
    // displays Hello World! on screen  
    cout<<"Hello World!";  
    return 0;  
}
```

Input Text pattern

```
string variablename;
```

```
getline( cin, variablename );
```

Input Text pattern

The cin object is an object of class istream

// C++ program to demonstrate the cin object

```
#include <iostream>
```

```
using namespace std;
```

```
// Driver Code
```

```
int main()
```

```
{
```

```
    string s;
```

```
// Take input using cin
```

```
cin >> s;
```

```
// Print output
```

```
cout << s;
```

```
return 0;
```

```
}
```

Challenges with Cin in C++

- If you use “cin” like this

```
string name;
```

```
cin >> name;
```

- Characters after space is not considered e.g. Fred Bloggs
 - Firstly it only reads “Fred”
 - Secondly “Bloggs” will be read by the next cin
- Hence, use getline()

```
string fullName;  
cout << "Type your full name: ";  
getline (cin, fullName);  
cout << "Your name is: " << fullName;  
  
// Type your full name: John Doe  
// Your name is: John Doe
```

Variables in C++

Syntax of declaring a variable in C++

```
data_type variable1_name = value1, variable2_name = value2;
```

```
int num1=20, num2=100;
```

Or

```
int num1,num2;
```

```
num1=20;
```

```
num2=100;
```

Types of Variables???

Difference between Global variable and Local Variable??

Global Variable and Local Variable

```
#include <iostream>
using namespace std;
// This is a global variable
char myVar = 'A';
char myFuncn() {
    // This is a local variable
    char myVar = 'B';
    return myVar;
}
int main()
{
    cout <<"Function call: "<< myFuncn()<<endl;
    cout <<"Value of myVar : "<< myVar<<endl;
    myVar='Z';
    cout <<"Function call: "<< myFuncn()<<endl;
    cout <<"Value of myVar: "<< myVar<<endl;
    return 0;
}
```

Global Variable and Local Variable

```
#include <iostream>
using namespace std;
// This is a global variable
char myVar = 'A';
char myFuncn() {
    // This is a local variable
    char myVar = 'B';
    return myVar;
}
int main()
{
    cout << "Funcn call: " << myFuncn() << endl;
    cout << "Value of myVar: " << myVar << endl;
    myVar = 'Z';
    cout << "Funcn call: " << myFuncn() << endl;
    cout << "Value of myVar: " << myVar << endl;
    return 0;
}
```

Executed 2nd

Executed 1st and 3rd

Executed 4th

OUTPUT:

Function call: B

Value of myVar : A

Function call: B

Value of myVar: Z

Exercise

1. Write a C++ program to get name, age and course as input and print the same in different lines
2. Write a program to double a given number

(Eg: Input: 5

Output:10)

Exercise 2

- Complete the following to **output** the numbers:

- *type* wholeNumber;
- Input wholeNumber
- `cout << ..... wholeNumber;`

- *type* totalCost;
- Input totalCost
- `cout << ????`;

- *type* countOfDogs;
- Input countOfDogs;
- Display “There are ?? dogs”

Write the code

- *type* peopleInStadium;
- Input peopleInStadium
- Display the contents of the variable like this “The number of people in the stadium is ??”

- *type* ropeLength = 4.56;
- *type* half = ropeLength / 2;
- Calculate and display half the length:
“Half the length is 2.28”

- *type* number1 = 5;
- *type* number2 = 11;
- Calculate and display the sum like this
“The sum of 5 + 11 = 16”
number1 + number2

} Try different values

Control Statements

- If, if..else-if statement
- Switch Case in C++
- For loop
- while loop
- do while loop
- Continue statement
- Break statement
- goto statement

If else Statement in C++

Decision making- Execute a block of statements only when a particular condition is met or not met.

four types of control statements

- a) if statement
- b) nested if statement
- c) if-else statement
- d) if-else-if statement

Program code

```
if (condition) {
```

```
// block of code to be executed if the condition is true
```

```
}
```

Example:

```
int x = 20;
```

```
int y = 18;
```

```
if (x > y) {
```

```
    cout << "x is greater than y";
```

```
}
```

If-else

```
if (condition) {
```

```
    // block of code to be executed if the condition is true
```

```
} else {
```

```
    // block of code to be executed if the condition is false
```

```
}
```

Try to write a piece of code demonstrating if-else

Arrays

- Store multiple values in a single variable.
- Example:

Declaration:

```
string cars[4];
```

To Insert Values:

```
string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};
```

Access the Elements of an Array

Program code:

```
int main() {  
    string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};  
    cout << cars[0];  
}
```

Exercise

Write a C++ program to display the third element in a given array of integers

Summary

- Topics covered
 - Classic 'Hello World' Example
 - Input/Output Syntax
 - Variable Declaration
 - Control Statements
 - Arrays
- Next Session
 - Classes and Objects in C++