

Getting started with C++

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

int main() {
    // Interesting code starts
    int principal;
    double interestRate;
    int numberOfYears;

    cout << "How much are you investing?\n";
    cin >> principal;
    cout << "What's the annual interest rate (%)\n";
    cin >> interestRate;
    cout << "How long for (years)\n";
    cin >> numberOfYears;

    double finalBalance =
        pow(1.0 + interestRate * 0.01, numberOfYears)
        * principal;
    double interest = finalBalance - principal;

    cout << "You will earn ";
    cout << interest;
    cout << "\n";
    /* Interesting code ends */
    return 0;
}
```

Comments

```
// Interesting code starts  
...  
/* Interesting code  
   ends */
```

Everything before and after these comments is *boiler plate*.

Declare variables

```
int principal;  
double interestRate;  
int numberOfYears;
```

Read input

```
cout << "How much are you investing?\n";  
cin >> principal;  
cout << "What is the annual interest rate (%)\n";  
cin >> interestRate;  
cout << "How long for (years)\n";  
cin >> numberOfYears;
```

Do some maths

```
double finalBalance =  
    pow( 1.0 + interestRate * 0.01, numberOfYears )  
    * principal;  
double interest = finalBalance - principal;
```

We are simultaneously declaring the type and assigning the value.

Print out the results

```
cout << "You will earn ";  
cout << interest;  
cout << "\n";
```

Spacing

- ▶ the ; character is used to separate statements.
- ▶ C++ doesn't care much how you space your file.
- ▶ You could put all the code discussed on one line.
- ▶ Use spacing and tabs for readability.

Compiled languages

- ▶ Computers don't speak C++ they speak “assembly language”, also known as “machine code”.
- ▶ The C language is very close to machine code.
- ▶ C++ is “compiled” directly into machine code.
- ▶ What you give your customers is a binary file not the *source code*.

Compiling and running

- ▶ Create a new Visual Studio project called InterestCalculator. Make a note of where it is saved.
- ▶ Set your project properties so that the SubSystem is Console. (Under **Project** → **InterestCalculator Properties...** → **Configuration Properties** → **Linker** → **System**).
- ▶ Add a new C++ source file called main.cpp
- ▶ Copy and paste the code into main.cpp
- ▶ Press CTRL + F5 to run.

Why so complex?

- ▶ For big programs it is convenient to group all your code into a project so you can set all properties in one place.
- ▶ C++ is designed for big projects not small ones.

What files were created?

- ▶ InterestCalculator/main.cpp
- ▶ Debug/InterestCalculator.exe
- ▶ InterestCalculator.sln
- ▶ Files used by Visual Studio.

Debug versus Release

- ▶ Release code is usually optimised for speed.
- ▶ Debug code is usually optimised for ease of debugging.

Dealing with errors

★ In the line

```
int principal;
```

remove the semi-colon and compile.

- ▶ What happens?
- ▶ How helpful is the message?
- ▶ Try enabling word-wrap.
- ▶ How accurate is the message?

Dealing with errors

- ★ Fix the code. Now repeat the exercise but remove the (symbol in the calculation of the final balance instead.

Dealing with errors

- ★ This time delete the { from the line

```
int main() {
```

- ▶ Is the last error message helpful?
- ▶ Is the first error message helpful?

Dealing with errors

- ★ This time replace the code

```
#include <iostream>
```

with

```
x  
#include <iostream>
```

What we have learned

- ▶ Fix the first error and then recompile. Later errors might be nonsense.
- ▶ Error messages try to be helpful but don't always succeed.
- ▶ If the compiler says there is an error in someone else's code, it is probably wrong. The error is in your code.
- ▶ Lots of error messages does not mean lots of errors, so don't panic.

A second C++ program

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

int main() {

    int mark;
    cout << "What was your average mark?\n";
    cin >> mark;

    if (mark >= 70) {
        cout << "Congratulations!\n";
        cout << "You got a distinction.\n";
    }
    else if (mark >= 60) {
        cout << "Well done!\n";
        cout << "You got a merit!\n";
    }
    else if (mark >= 50) {
        cout << "You passed.\n";
    }
    else {
        cout << "You failed :-(\n";
    }
    return 0;
}
```

The interesting bit

```
int mark;
cout << "What was your average mark?\n";
cin >> mark;

if (mark>=70) {
    cout << "Congratulations!\n";
    cout << "You got a distinction.\n";
} else if (mark>=60) {
    cout << "Well done!\n";
    cout << "You got a merit!\n";
} else if (mark>=50) {
    cout << "You passed.\n";
} else {
    cout << "You failed :-(\n";
}
return 0;
}
```