Introduction to C++

Dr Ian Cornelius

Hello

Hello (1)

Learning Outcomes

1. Understand the background and the process of working with C++
2. Demonstrate the ability to work with C++ using an integrated development environment

Overview of C++

Overview of C++ (1)

- A cross-platform language and is often used to create high-performance applications
- Provides programmers with a high level of control over system resources and memory
- Four major revisions to the language in 2011, 2014, 2017 and 2020
  - a fifth is on the way 2023...
  - referred to as C++11, C++14, C++17 and C++20

Overview of C++ (2)

History of C++

- Developed by Bjarne Stroustrup in 1983 as an extension to the C programming language
  - development has begun four years prior to release in 1979
  - first known as *C with Classes*
- First version of C++ released in 1983, and renamed from *C with Classes*
- Major versions released through the years:
  - C++98, C++11, C++14, C++17, and C++20

Overview of C++ (3)

Features of C++

- C++ is an object-oriented language
- Declaration of **global** variables is not allowed
- Operators `new` and `delete` are used for memory allocation and deallocation
- C++ is **machine independent**, meaning it is cross-platform compatible

Overview of C++ (4)

Advantages of C++

- C++ can be used for a wide variety of tasks, such as:
  - General System Applications
  - Video Game Development
  - Applications for Servers
- A mature language which is clear and consists of a large community
- Aids developers to create applications with *re-usability* and *readability* in mind
- Known as a mult-paradigm degree

Overview of C++ (5)

Disadvantages of C++

- Misusing pointers, whereby the system may crash or behave in a weird manner
- The language can often be complex

Using C++

Using C++ (1)

- Using **modern** C++ for this module, C++20
  - although C++23 is on the horizon
  - it is cross-platform and works across all major operating systems
- Requires an installation of a C++ compiler
  - Windows: Download the Executable Here
  - Linux: `sudo apt install build-essential`
  - macOS: `xcode-select --install`

Using C++ (2)

macOS, Linux and Windows

- Typing `g++` into the terminal window will call the C++ compiler
  - i.e. `g++ file_name.cpp -o file_name`
- You are more than likely going to compile your code using the integrated development environment (IDE)
  - the IDE for this module uses `cmake` to build your C++ application

Using C++ (3)

Integrated Development Environment

- Supported IDE: **JetBrains CLion**
  - Apply for an Educational Licence
  - Download JetBrains CLion
- Features:
  - Debugging
  - Code Refactoring and Profiling
  - Version Control Integration
  - Compilation of C++ source files
- **Note**, you are still expected to learn how to do things via the terminal/command-line
  - especially with the version control tool and source-code compilation

Using C++ (4)

Writing your C++ Code

- Line 1: starts by including a module for managing input and output
  - enables writing output to the terminal window, e.g. `cout`
- Line 3: creates a `main` function to execute the script
  - this is the *entry point* of the script
- Line 4: prints a string using the `cout` function from the `iostream` module

```
#include <iostream>

int main() {
   std::cout << "Hello 5062CEM!\n";
   return 0;
}
```

Using C++ (5)

Compiling your C++ Code

- Requires using the `g++` command that was introduced earlier
- Compilation can be achieved using the following command:
  - `$ g++ main.cpp -o main`
- Remember, the dollar (`$`) character indicates this is a terminal command/script
- Executing the binary file can be achieved with the following command:
  - `$ ./main`

```cpp
#include <iostream>

int main() {
   std::cout << "Hello 5062CEM!\n";
   return 0;
}
```

```
Hello 5062CEM!
```

Goodbye

Goodbye (1)

Questions and Support

- Questions? Post them on the **Community Page** on Aula
- Additional Support? Visit the Module Support Page
- Contact Details:
  - Dr Ian Cornelius, ab6459@coventry.ac.uk