COMP110: Principles of Computing

Transition to C++ I

Learning outcomes

By the end of this session you will

- Understand a thing
- Understand another thing
- Be convinced that LaTeX makes better-looking slides than PowerPoint

FALMOUTH
UNIVERSITY

Your first C++ program

Project setup

- Open **Visual Studio 2015** from the Start menu
- Click **New Project**
- Choose **Templates** $\rightarrow$ **Visual C++** $\rightarrow$ **Win32** $\rightarrow$ **Win32 Console Application**
- Choose an appropriate name and location, and click **OK**
- Click **Finish**
- When asked about source control, click **Cancel**

The code

- Edit ⟨`YourApplicationName`⟩`.cpp` to match the following:

```
// ConsoleApplication1.cpp : Defines the entry point for the console application.

#include "stdafx.h"
#include <iostream>

int main()
{
    std::cout << "Hello, world!" << std::endl;
    return 0;
}
```

- Click **Debug** → **Start Without Debugging**, or press **Ctrl + F5**

Comments

```
// ConsoleApplication1.cpp : Defines the entry point ↩
    for the console application.
```

- `//` denotes a single-line comment
- Equivalent of `#` in Python
- ↩ denotes a line too long to fit on the slide — in your program this should be a single line

The #include directive

```
#include "stdafx.h"
#include <iostream>
```

- `#include` imports definitions from a **header file**
- Similar to `import` in Python
- `#include "..."` (quotes) is used for headers in the current project
- `#include <...>` (angle brackets) is used for external libraries

Entry point

```
int main()
```

- All code must be inside a function
- The **entry point** of an application is (almost) always named `main`
- `int` means the function returns a value of integer type
- `()` means the function takes no parameters

Blocks and semicolons

```
{
    ...;
    ...;
}
```

- Curly braces are used to denote blocks
- All statements in C++ end with a semicolon ;
- Unlike Python, C++ ignores whitespace (indentation and line breaks)
- ... but whitespace is important for readability, so use it anyway

Writing to the console

```
std::cout << "Hello, world!" << std::endl;
```

- Equivalent of Python's `print` statement
- `std` is the **namespace** containing most of the C++ standard library
- `std::cout` is the console output stream
- `std::endl` is the end-of-line character
- To use `std::cout` and `std::endl`, it is necessary to `#include <iostream>`
- `<<` is the **insertion operator** — used to write values to a stream

Exit code

```
    return 0;
```

- Returning 0 from `main` tells the OS that the program completed successfully

FALMOUTH
UNIVERSITY

The compiler

The build process