Introduction to Coding and c++

Thomas Merkh
tmerkh55@gmail.com

Section 2, Week 1

April 2, 2020
Overview

1. What is computer programming anyway?

2. First Milestone: Build and Run a simple program in Visual Studio

3. Breaking Down “Hello World”
What is programming, and why bother

- Programming amounts to custom instructions you provide to computer(s) to *do something*

- Examples: Create a movie database, approximate the solution to a mathematical equation, create a website, develop an animated game, or just have speakers play the same song on repeat

- To talk to a computer, you need to speak a computer language (C++ will be the language for this class)

- In addition to knowing the syntax, one must know a bit of basic algorithms and “what’s going on under the hood”
How do I get my computer to listen to my instructions?

You will need certain things installed on your computer for it to know how to read c++ instructions. One must have installed

- A text editor (notebook, sublime, MS word, anything you can type in)
- A c++ compiler (this *basically* translates the code you write into ‘1’s and ‘0’s)
- c++ libraries (these define the c++ syntax that your computer will understand)

Seems like a lot? What if these different things aren’t playing nice?

Relevant tangent: Ex-roommate “Make a movie database” story.
How do I get my computer to listen to my instructions?

For this class, we will use an IDE “Integrated development environment”. These are all-in-one solutions so you do not need to worry about compatibility issues.

- **Pro**: All-in-one solution, no need to worry about installing the correct components individually
- **Neutral**: IDEs have a million features, some are useful (like debuggers)
- **Con**: There is a significant learning curve
- **Con**: IDEs are operating system dependent

This class will use the IDE “Visual Studio” for Microsoft Windows.
#include <iostream>

```cpp
int main() {
    std::cout << "Hello remote Earthlings" << std::endl;
}
```

Once everything is set up and working, you will likely just press F5 to run (or something similar). However, what is actually going on?

Suppose that you have a c++ file, Foo.cpp. When you hit F5, the computer does:

- Compiling: Foo.cpp → Foo.obj
- Linking: Foo.obj + “c++ libraries/system routines” → Foo.exe
- Running: Foo.exe

See next slide for visualization of this process.
The Flow

Program Code

You write this!

Compiler

Many options like runtime optimization

Linker

Static or Dynamic

Executable file

A stand-alone program you can run
What is computer programming anyway?

First Milestone: Build and Run a simple program in Visual Studio

Breaking Down “Hello World”
Where To Begin?

First Milestone: Build and Run a simple program in Visual Studio

Where to begin? For this, we need to:

- Have Visual Studio properly installed
- Learn enough syntax to give the instructions to print “Hello World”
- Know how to tell Visual Studio “Compile and run this .cpp file”

If this sounds trivial, perhaps you’ve coded using VS before. If you haven’t then you’ll see how much is involved!
Step 1: Installing Visual Studio

Visit https://visualstudio.microsoft.com/downloads/, click on “Free download” under Community.

- Run the executable file you downloaded
- It will ask you to choose which workloads you’d like to install
- Check the “Desktop development with c++” box
- **Warning**: VS can be HUGE if you choose to install other workloads
- Click install, and I’d recommend installing to the default locations on your computer

*Tip:* Wired internet connection is nice for downloading many Gbs!
Step 2: Create Your First Project

In Visual Studio, there are *projects* and *solutions*.

- Solutions are large containers which can hold many individual projects (usually all related projects)
- Projects are comparable to folders which hold all the files for a single assignment
Step 2: Create Your First Project

In Visual Studio, there are *projects* and *solutions*.

- Solutions are large containers which can hold many individual projects (usually all related projects)
- Projects are comparable to folders which hold all the files for a single assignment

To start your first project, open up VS and click “Create new project”. Then

- You may have choices, if so, choose “Empty Project” and choose a location for it in your file system
- An empty VS screen will result, with your current *project* open
- To add a .cpp file, right-click on your project name in the side bar, go to *Add, New Item*
- Choose .cpp file, it will likely name the file “Source.cpp” for you
Figure: The Visual Studio work space, adding a file to the project
Step 3: Copy my source code and compile

Remember my program?

```cpp
#include <iostream>

int main() {
    std::cout << "'Hello remote Earthlings'" << std::endl;
}
```

Copy this into that blank .cpp file and save it. To run it, try one or both of the following:

- Hit F5 (or Fn+F5 for some laptops)
- Go to “Build”, click “Build Solution”, then go to “Debug”, click “Start Debugging”

Tip: Copying quotation marks from powerpoint slides for some reason doesn’t work well, so fix that in your code.
Congratulations! You’ve run your first .cpp code using Visual Studio

Questions?
1. What is computer programming anyway?

2. First Milestone: Build and Run a simple program in Visual Studio

3. Breaking Down “Hello World”
Analyzing “Hello World”

Let’s look at

```cpp
#include <iostream>

int main() {
    std::cout << "Hello remote Earthlings" << std::endl;
}
```

What are the essential components? What do we notice that looks strange?
Analyzing “Hello World”

Let’s look at

```cpp
#include <iostream>

int main() {
    std::cout << "Hello remote Earthlings" << std::endl;
}
```

What are the essential components? What do we notice that looks strange?

- Include statement(s)
- integer main?
- std:: symbols
- The infamous semi-colon
- The curly brackets to close the main function
Analyzing “Hello World”

- Include statement - Tells the compiler to include pre-written libraries containing the “cout” object we use later.

- int main - All functions in C++ are objects. They have a type, and the main function is (always) and integer. This will become more clear later.

- std:: - This tells the compiler where to look for the “cout” object. “Look in the std library for that command”.

- semi-colon - They tell the compiler where a line ends. A very important thing to know!

- The curly braces - these will enclose scopes, another concept which will be taught later.
Until next time, I would suggest searching around Visual Studio and learning how it works. There are many video tutorials online. Next time we will be discussing

- Variables - Entirely fundamental
- Types - every object in C++ has a type
- Input/Output - it's hard to do much without seeing an output printed to the screen

See you then!