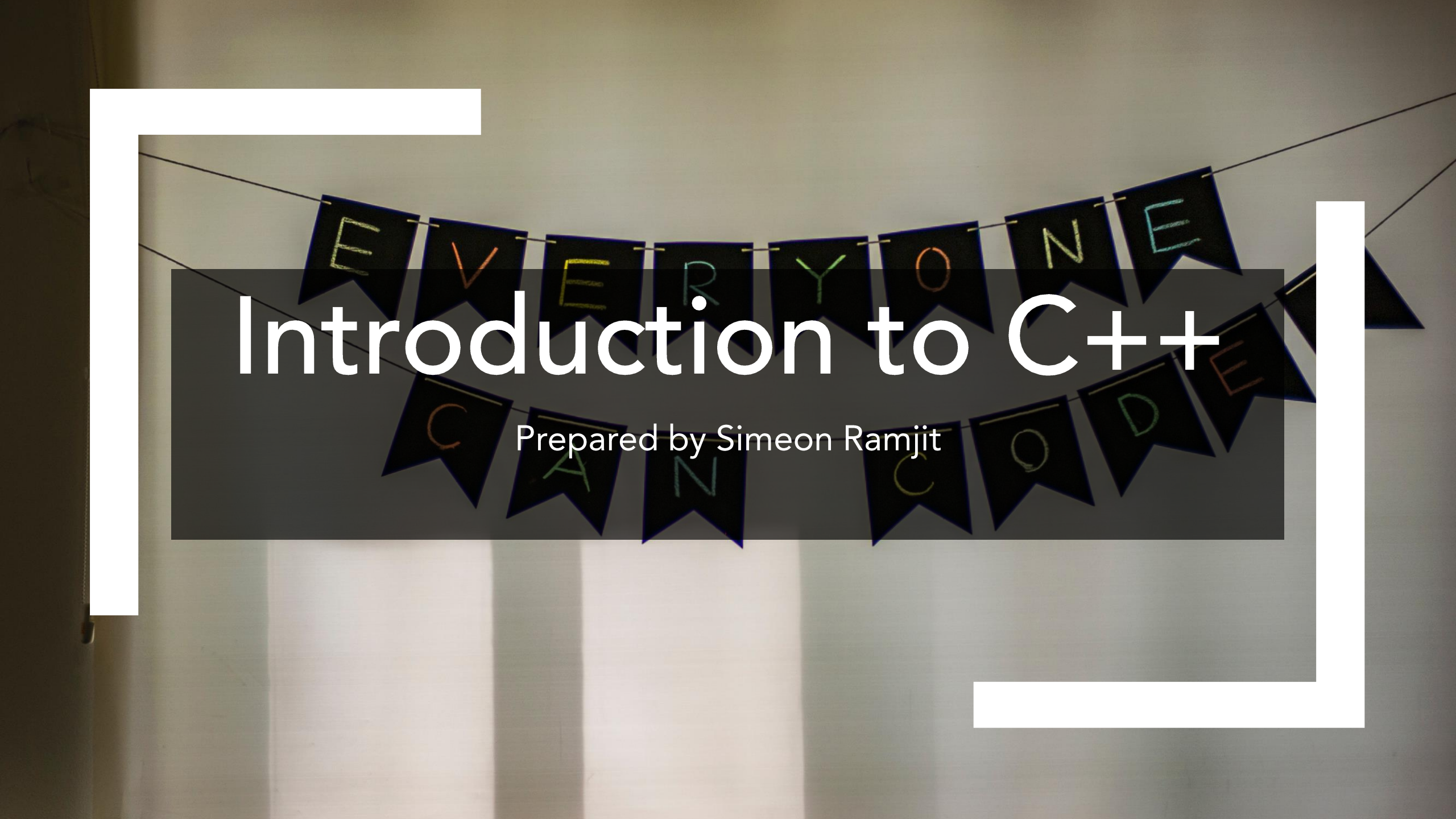

Introduction to C++

Prepared by Simeon Ramjit

What is C++

- C++ (C Plus Plus) is one of many **programming** languages that exist.
- It is an extension of the C language, mainly used in embedded systems, and created by Bjarne Stroustrup in 1979 at Bell Labs. It was originally called “C with classes”
- The alternative to a programming language is a mark-up or scripting language, like HTML (Hyper Text Mark-up Language) which is used to write the structure of webpages.
- Mark-up languages are **not** used to write programs

What is C++

- C++ has a standard as well. There were 3 major updates
  - 2011 to C++11
  - 2014 to C++14
  - 2017 to C++17
- It is one of the few languages that give developers high level control over computing resources and memory (RAM).
- This control is abstracted away in languages like Python

Getting Started - C++

- You will need an Integrated Development Environment (IDE) to write code in
- For Windows users, Visual Studio Community is usually the go-to
  - Link to install: https://youtu.be/iV1k7nwfM94
- For MacOSX , VS Community isn't supported for C++ development. You'll need another IDE or in this case a code editor like VS Code
  - Link to install and project creation: https://youtu.be/3tlDLNUbU98
- Files that contain C++ code usually end with .h (header files) or .cpp

Getting Started - C++

- Much like any language C++ has it's own set of rules and way that the language is written (syntax)
- If spelling errors or errors that violate the syntax rules occur in your code, they are referred to as syntax errors
- There are quite a few different types that we will get to later like logical and runtime

Getting Started – VS Community

- Let's create a project - Windows users:
  1. Open visual studio Community
  2. Create a new project
  3. Empty project (if this does not come up see next slide) > Next
  4. Enter a project name (if you want, change the location of the project folder by clicking the three dots to the right of the location bar)
  5. Check "Place solution and project in the same directory"
  6. Click create - your project should now be loaded with the Solution Explorer on the right. If the SE isn't open, press Ctrl+Alt+L to load it
  7. In the SE, right click "Source Files" > Add > New Item > C++ File (.cpp)
  8. Give it any name you want then click "add"

- NB: Without any source files added, the project will fail to build. If the wrong program is building make sure the correct source files are added.

Getting Started - VS Community

- If you did not see any C++ options when in the project creation:
  1. Open Visual Studio Community
  2. Click “Continue without code”
  3. Look at the toolbar (up top) and click Tools > Get Tools and Features
  4. Say yes and let it load
  5. Under the tab “Workloads” look for “Desktop and Mobile” then make sure “Desktop Development with C++” is checked and click install

Getting Started - VS Community

- MacOS X users, your getting started is included in the video tutorial.
  - *You can use an online C++ shell at*

    https://www.onlinegdb.com/online_c++_compiler
- VS Community, remember these two commands
  - Build (Ctrl + Shift + B)
  - Run **without** debugging (Ctrl + F5)

Your First Program! 🎉

- Write in the code below in the .cpp file you created. Then build and run without debugging. Alternatively, just run without debugging (includes the build step)

```
#include<iostream>

int main() {
  //Prints Hello World! to the screen
  std::cout << "Hello World!" << std::endl;
  return 0;
}
```

Whoa, that's a lot to unpack 🤔

- Almost all lines have a semicolon to end them. In C++, a ';' is used to denote the end of a **statement.**
- Line 1 – A pre-processor directive that allows you to load that header library for use in your program. Libraries are used to add functionalities to your program without having to rewrite code from basics. In this case the use of the `<iostream> (input-output stream)` enables us to **stream** information, mainly text and runtime generated *variables*, in and out of the program.
- Line 2 – A blank line. You can have blank lines, they are ignored at build time. Blank lines can be used to add 'breathing space' so code becomes readable

Whoa, that's a lot to unpack 🤔

- Line 3 – `int main(){}`
  - A function of type integer, we'll get to functions later. C++ uses this as the entry point for execution. This means the program starts execution from just after the first parenthesis (curly brace) till just before `return 0;`. If there is no main function there is no execution. The curly braces also define the scope of the function `main`.
- Line 4 – That is a single line comment. Single line comments are created with //. They are useful to leave notes as you go along (good practice) or to comment out code temporarily. Comments are ignored at build time. Multiline comments begin with /* and end with */

Whoa, that's a lot to unpack 🤔

- Line 5 – `return 0;`
  - Remember we said that **`main`** was a function of type integer (**`int`**)? Functions that have a type usually* have to give back a value after execution i.e. they have to return a value. The main function is special, it always returns a value of 0. This value indicates to your operating system that the program has finished execution and it did so with no errors. This allows the operating system to reallocate resources that would have been used to run your program. Only one **`main`** can exist for a program.
- Do not use blocking code if your terminal window closes and pops down when run. See end of presentation for solution to that.
  - ☠️`SYSTEM("PAUSE");`☠️
  - ☠️`getchar();`☠️

Your Second Program! 🎉

- Write in the code below in the .cpp file you created. Then build and run without debugging. Alternatively, just run without debugging (includes the build step)

```
#include<iostream>
int main() {
 int number = 0; //Declares a variable of type integer
 std::cout << "Enter a number: ";
 std::cin >> number; //Note the stream direction changes
 std::cout << "The number is: " << number << std::endl;
 return 0;
}
```

And that’s a wrap!

- Well, almost. You know how to write a basic program in C++ but you also need to get an idea of how you go from a programming language to the binary that a computer can understand
- This is a fantastic video explaining the process, watch it instead of Netflix plz and no it’s not a random Indian guy on YouTube.
- How do computers read code? – Frame of Essence
  - https://youtu.be/QXjU9qTsYCc

Visual Studio Community Fixes

- If the terminal window closes as soon as the program finishes execution.
  - Project > “yourproject” properties > Linker > System > ‘Subsystem’ – make sure it’s set to console.
- LNK1168
  - Make sure your terminal window is closed before running your program again
  - If that doesn’t work: Project > “yourproject” properties > Linker > General > ‘Enable Incremental Linking’ – make sure it’s set to no.

Questions?

simeon.ramjit@sta.uwi.edu

github.com/simeon9696/programmingworkshop