Week 0 Introduction

Tyler Swann




Welcome

Welcome to High Performance Programming Team. Over the course of this workshop series, you will learn to use the C++ programming language and create high performance software.

Agenda

- Introductions
- This week's meetup (agenda)
- Series overview & timeline
- Resources
- Content
- Discussion

Meeting the Team

Tyler Swann

Manager

Name

Title

Name

Title

Name

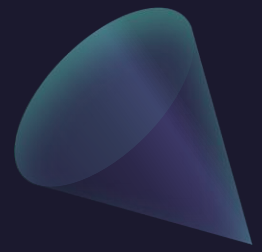
Title

Series Overview

WHAT WE HOPE TO TEACH

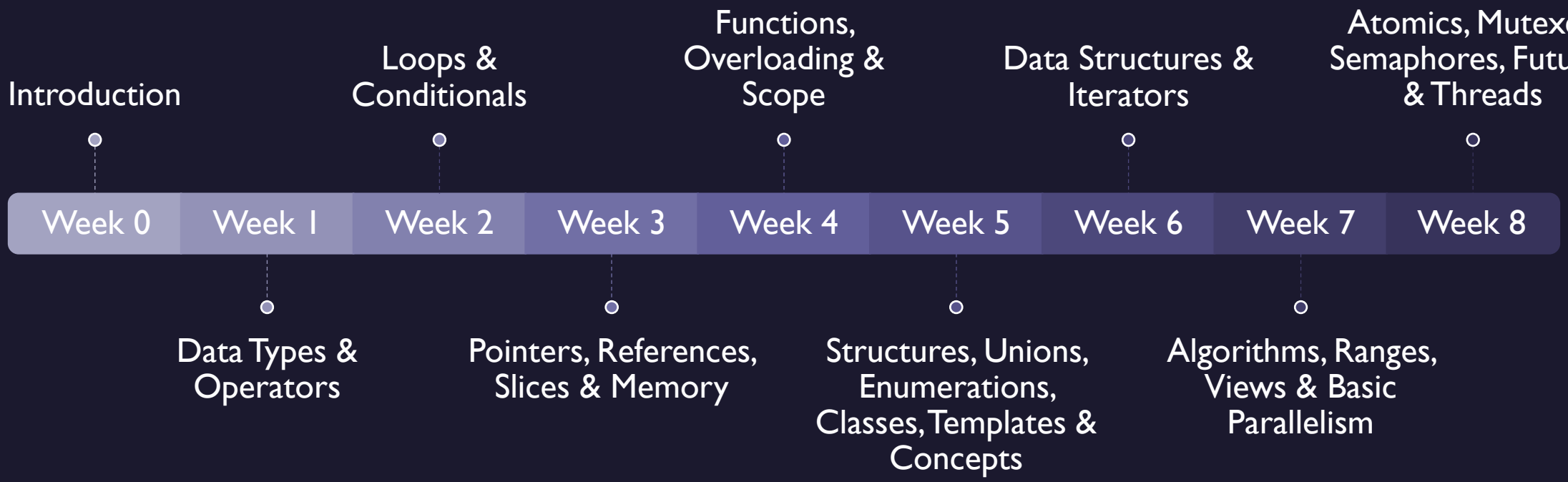
- A good, foundational understanding of programming techniques.
- Knowledge of the C++ programming language.
- Basics of computer software architecture
- Ownership and memory concepts
- Generic programming techniques
- Algorithm intuition
- Beginners guide to parallelism primitives and techniques

TECHNOLOGY YOU WILL USE



- C++
- Markdown
- Bash
- Git + GitHub
- Makefiles
- bpt
- VSCode

Timeline


Introduction
Loops & Conditionals
Functions, Overloading & Scope
Data Structures & Iterators
Atomics, Mutexes, Semaphores, Futures & Threads
Week 0
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Data Types & Operators
Pointers, References, Slices & Memory
Structures, Unions, Enumerations, Classes, Templates & Concepts
Algorithms, Ranges, Views & Basic Parallelism

Installing Software

Week 0

Git & GitHub

ABOUT

- Git is a software control management (SCM) tool that helps to manage software projects and keep a record or the software's history.
- Projects are managed using repositories, which contains information about the branches, history and contributors of a project.
- GitHub is a website that can host remote (online) repositories.
- You can `clone` and `fork` repositories from GitHub allowing you to use and contribute to them.

SETUP GITHUB

- Create a GitHub account. Use a personal email as you will most likely want to keep your account after university.
- We will collect your Username + Email used to add you to the MDN GitHub organization and to add you to the relevant team.
- Keep note of your username and email as you will need it later.

HPP Repository & MDN Organization

HPP REPOSITORY

- For this series, all content will be available on a repository on GitHub called HPP. It is highly recommended you bookmark or star this repository so you can easily access it.
- The weekly slides (*.pptx and *.pdf formats), code examples, tasks, instructions and content are hosted here.
- The instructions for each week will also be hosted on the repository in a read-only format. This should make copying commands and code snippets easier.
- E.g. All commands and instructions for installing all the software this week are in `content/week0/tasks/README.md`. These slides are non-comprehensive so follow along with the GitHub.
- Link: https://github.com/MonashDeepNeuron/HPP

MDN ORGANIZATION

- The MDN organization hosts the projects and resources for MDN projects and members.
- Permissions and access are managed through teams.
- Each team has a private discussion, project board and different repository access.

Windows Terminal and WSL

WINDOWS TERMINAL

- Windows Terminal is an application by Microsoft that is able to host multiple shells.
- Allows for a better developer experience as much of C++ development is in the terminal.

WSL

- WSL is a native Linux Kernel that can run alongside Windows.
- It also has access to your computers main filesystem along cross platform development.

Homebrew and VSCode

HOMEBREW

- Homebrew is a cross-platform package manager.
- It allows us to install system packages without needing the systems package manager, such as apt.

VSCODE

- VSCode is a text editor
- It provides high utility through its extension marketplace so you can customize your developer tool kit as needed.

Other Software

- wget – Online asset retrieval
- cURL – Online asset retrieval
- make – C build recipe tool
- Cmake – Cross platform recipe builder
- bpt – Build and packaging tool for C++
- GCC – GNU Compiler Collection
- LLVM – Toolchain for Clang compiler
- GDB – GNU Debugger
- g++ and clang++ compilers
- Compiler Explorer – Online C++ Compiler

Hello World

Week 0

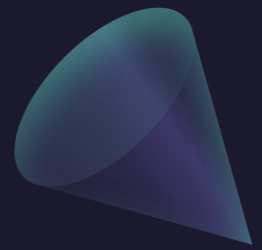
Introducing C++

C++ IS A _ LANGUAGE:

- High level
- General purpose
- Strongly typed
- Statically typed
- Compiled
- Multiparadigm

PARADIGMS

- Procedural
- Imperative
- Functional
- Declarative
- Object Oriented



Hello World

IT IS ALIVE!!

- A "Hello World" program is a simple program that;
    1. acts as a sanity check that everything is working.
    2. Showcases the basic syntax and structure of the language.
- It may seem crude to get a good idea of a language and to judge it based off a "Hello World" program in that language; especially if its your first, second or even third language but it can clue you in to what to expect.
- You should be able to copy the code to the right directly into VSCode, although it is encouraged to write by hand to get use to the grammar and structure.
- In the next slides we will break down the various components of this small piece of code.

THE CODE

```
// This is a comment, these are ignored by
the compiler

#include <iostream>

auto main () -> int
{
    std::cout << "Hello World!" << std::endl;

    return 0;
}
```

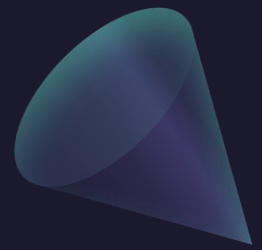
A Closer Look

Comments allow you to document you code

This is the main function. When your code executes, it starts here. Main returns an integer indicating the status of the program to the OS. The functions contents are contained in the pair of braces ({})

Comments allow you to document you code

This is a pre-processor directive. `#include` copies the contents of the file between the <> into the current file



```
// This is a comment
/// These are ignored by the compiler

#include <iostream>

auto main () -> int
{
    std::cout << "Hello World!" << std::endl;

    return 0;
}
```

`std::cout` is the character output stream. The << puts the string (between the "") to the stream. `std::endl` puts a newline character (`'\n'`) at the end of the stream and flushes `std::cout` buffer. Expressions must end with a semicolon (;)

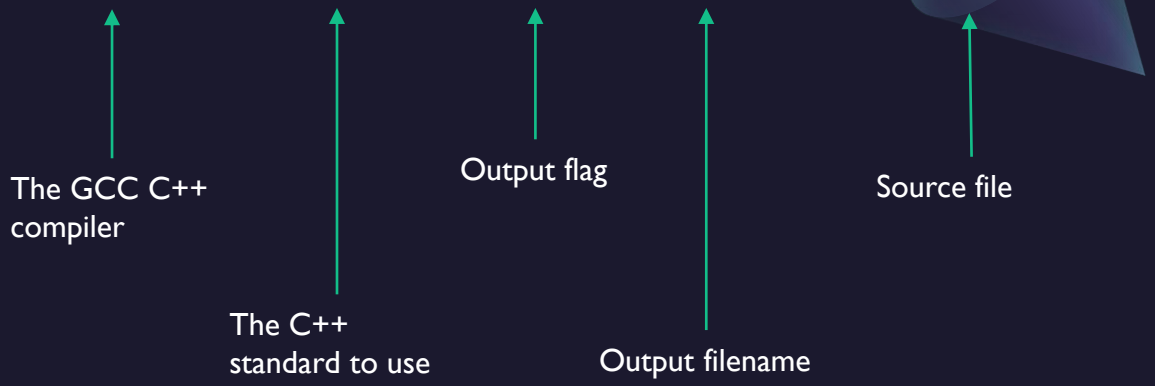
Compiling and Running

THE COMPILER

- Reading code isn't very interesting, especially when its as trivial as "Hello World".
- What is more interesting is seeing what is does. But first we need to compile the program.
- The compiler turns out source files into machine code that is able to run on our device.

THE COMMAND

```
g++-12 -std=c++20 -o build/hello hello.cxx
```



To run the code use:

```
./build/hello
```

Compiler Explorer and cppreference

COMPILER EXPLORER

- Compiler Explorer (Godbolt) is an online compiler for C++
- It allows you to test small C++ code snippets on different compilers, architectures and standards of C++.
- It also allows you to share permalink to C++ instances on Godbolt (see here).

CPPREFERENCE

- cppreference is an online C++ reference.
- It breaks down the C++ language and Standard Library features down into the individual sub-libraries an allows for up-to-date, per-standard references to all of C++.
- It is verbose at first, but it gives a good overview of everything you have available to you in the language, algorithmic complexities, defect reports and implementation details about every component.
- We will be using it extensively to showcase many C++ features

Discussion

- Any questions?
- Need help?
- Open discussion.
- Concerns?

Next Week

- Static Type System
- Primitive Data Types
- Variables
- Qualifiers
- Operators

Summary

Today you setup your device for developing C++. You got your first introduction to C++ and learnt a little about the type of things you can accomplish with C++.

Thank You

Tyler Swann

https://github.com/MonashDeepNeuron/HPP