ual: creative computing institute

Week 3

C++

Day 1	Tuesday 13-Sep-22	9am -10am + 4pm-6pm
Day 2	Wednesday 14-Sep-22	9am -10am + 4pm-6pm
Day 3	Thursday 15-Sep-22	9am -10am + 4pm-6pm

Date

Jasper Zheng

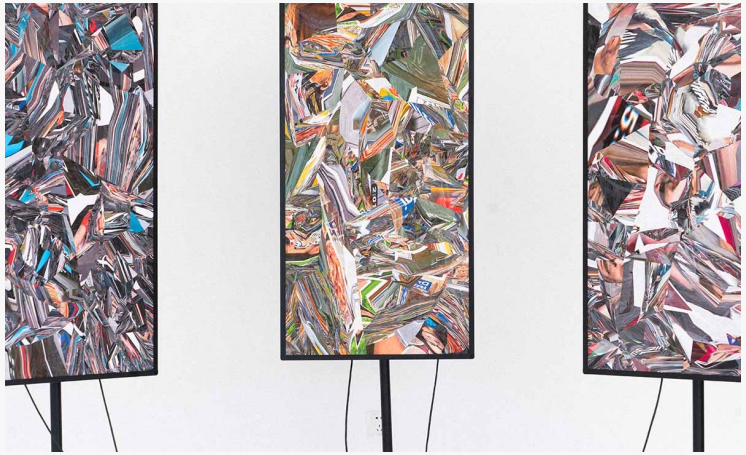
About Jasper:

Hi, my name is Jasper Zheng

I was raised in Beijing, studied at Computer Science Department in University of Liverpool (2017-21), currently studying creative computing at Creative Computing Insititute, UAL.

I use codes and programmes to create works, find the balance between machine autonomous and human creativity. Recently, I'm mostly interested in creative applications of artificial intelligence.

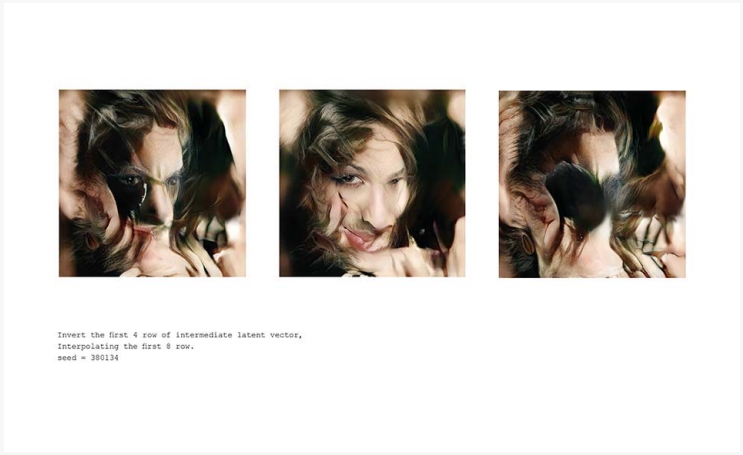
https://alaskawinter.cc/



The Third Dashboard (2019)

A data-reactive experiment tracks over the recommendation engine on YouTube, simulates perceived surroundings inside a social network.

https://alaskawinter.cc/works/the-third-dashboard



Manipulated Network (2022)

Collection of quirks and oddities generated from corrupted generative models.

https://github.com/jasper-zheng/manipulated-network

↑↑↑% (2021)

Aestheticise the chaotic dynamics in physical world.

https://alaskawinter.cc/works/up-percent


Lyra Starter Game - UE Game Samples

"To better communicate to our computers **what exactly it is we want them to do**, we've developed C++ to make the communication process easier."

C++ official documentation

In the next 3 days...

1

Setting up environment

Basic syntax / building blocks / inputs

Build a calculator

2

Array and loops

Control flows

2D arrays and nested loops

Build a Tic-Tac-Toe game

3

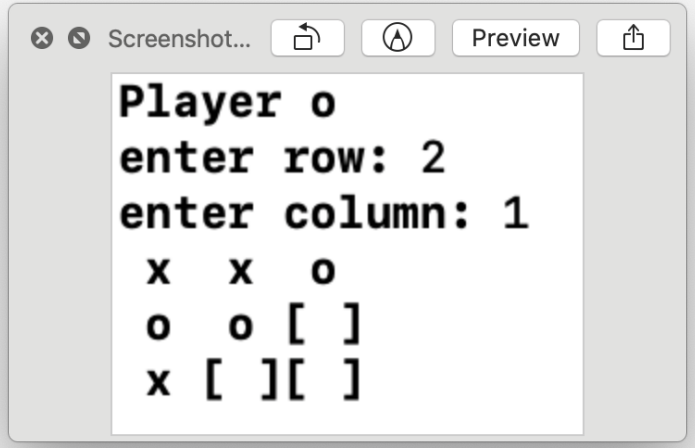
Object-oriented programming (a beautiful way to manage your program)

In the next 3 days...

Beside the syntax / operations / functions etc., another critical aspect in computing is computational thinking.

That is, how to design and program simple processes? How do we map real-world behaviour/interaction into a program? How do we present a solution in a way that both computer, and human can understand.

Each day we will take a simple problem (e.g. a calculator, a X and O game), utilise different computational techniques, solve them by a C++ program.



How you will/may use C++ at CCI

You will explore and experiment with advanced techniques for computation design, interactivity, embedded systems. You will have a wide range of choices regarding platforms, software, hardware, etc.

You may also work with Unreal Engine / Unity to create playable digital experiences.

The MODULAR Handbook

In which Units will you encounter the language in?

- Coding One: Advanced Creative Coding
- Coding Two: Advanced Frameworks
- Creative Making: Advanced Visualisation and Computational Environments

What are some examples of the work you will be expected to make ?

- Immersive experiences (VR/AR)
- Games / interactive systems
- Mobile / desktop applications

Link to Handbook: https://www.arts.ac.uk/__data/assets/pdf_file/0016/310723/MA_MSc-Computing-and-Creative-Industry-Modular-Book-of-Units-1.pdf

ual: creative computing institute

C++ Day 1

Introduction and "Hello World"

Date

Jasper Zheng

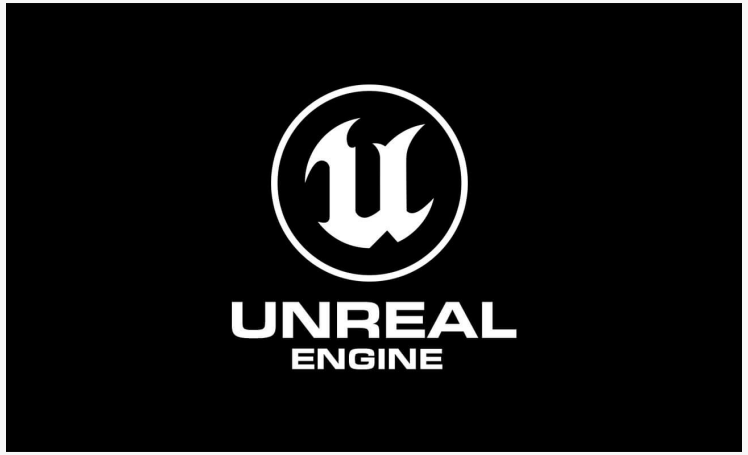
What is C++

As an extension to the C language, C++ gives programmers a high level of control over system resources and memory.

Strengths	Cons
Known for performance Useful for scalable applications Useful for game programming Low CPU and memory usage	Complex syntax and file systems Compiling a program can be tricky

Unity (Game Engine)

Language: C# (C-based language very similar to C++)

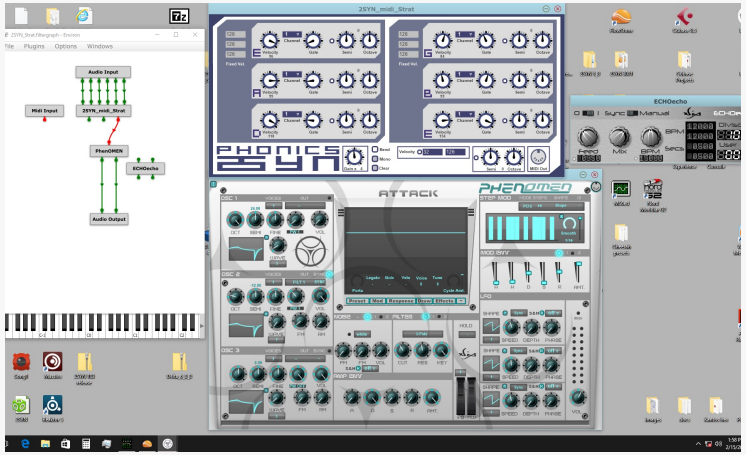


Unreal Engine (Game Engine)

Language: C++

Xcode (iOS App Dev)

Language: Swift (Application programming language extended from C++)



JUCE (Audio/MIDI plugin)

Language: C++

OpenFrameworks (Creative Coding)

Language: C++

What C++ is not ideal for:

- Online, networked environment (consider using JavaScript?)
- Machine learning / deep learning tasks (consider Python to start with).

Before we start...

Compiler vs. Interpreter

Compiled Language

A compiler takes entire program, converts it into machine code, then executed by the machine.

- Takes a compilation stage
- But usually run faster after compiled
- A single error would prevent the code from compiling.

C++ is a compiled language

Interpreted Language

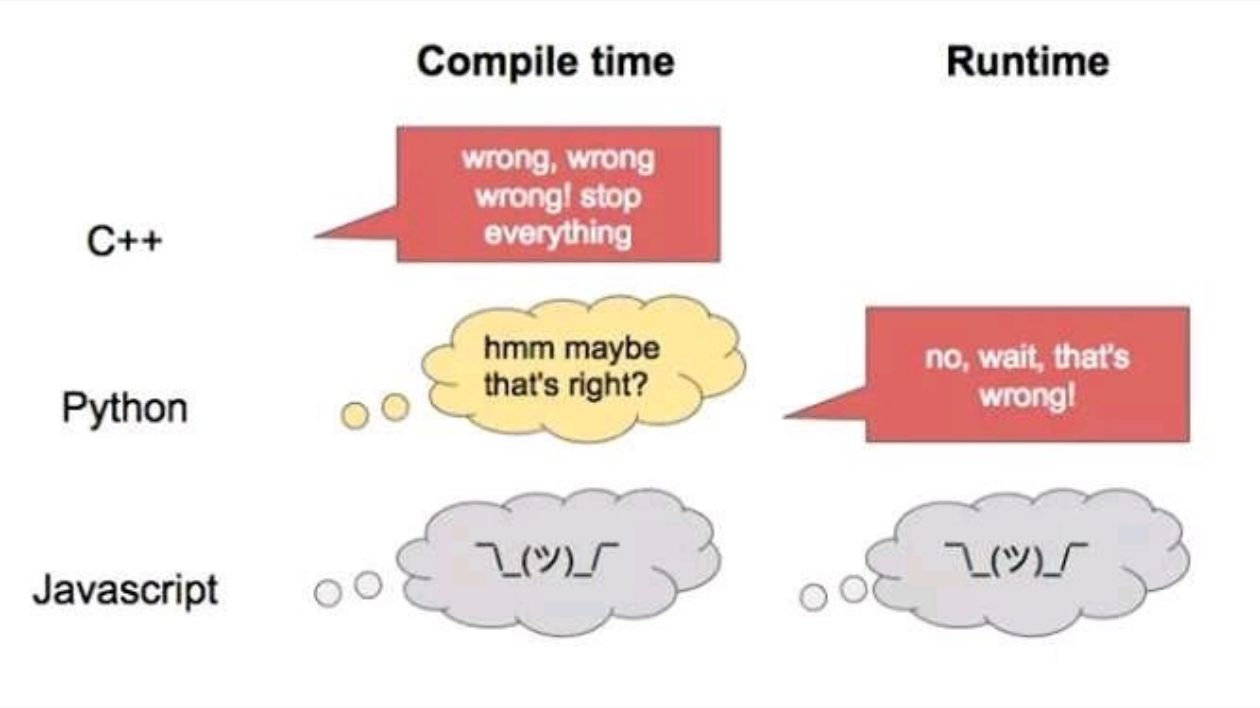
An interpreter directly executes instructions written in a script, line by line, without previously compiling.

- Easier to implement
- Can execute code "on the fly"
- Debugging occurs at run-time

JavaScript, Python are interpreted languages

Before we start...

Compiler vs. Interpreter



Source https://medium.com/@simonravi/a-quick-look-into-javascript-python-go-and-c-6878c8ee0dde

Before we start...

Statically-typed vs. Dynamically-typed

Statically-typed

The type of a variable is known at compile time

- We specify the data type of each variable
- And it's fixed after we declared them

```
int my_number = 10;
string my_number_as_text = "Ten";
```

C++ is a statically-typed language

Dynamically-typed

The type of a variable is checked during run-time

- We don't need to declare the data types of variables
- The type of a variable can be changed

```
my_number = 10
my_number = 'ten'
```

JavaScript, Python are dynamically-typed languages

Before we start...

Recap: Data Types

You may have already seen different types of data in Python / JS (whole numbers, fractional numbers, text, true and false...),

Read the chapter on variables, see how they are defined in C++: https://www.w3schools.com/cpp/cpp_variables.asp

- `int` - stores integers (whole numbers), without decimals, such as 123 or -123
- `double` - stores floating point numbers, with decimals, such as 19.99 or -19.99
- `char` - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
- `string` - stores text, such as "Hello World". String values are surrounded by double quotes
- `bool` - stores values with two states: true or false

Before we start...

An Example C++ Program

`#include <iostream>`

We use `#include` to tell the compiler which file, header, or library to be included in the program. Here we are including the `iostream` library, which contains essential functions that we are going to use in the hello world program.

`using namespace std;`

We are telling the compiler to use everything under the `std` (standard) namespace.

`int main()`

Every C++ program must have a main function, the compiler will look for the `main()` function to execute your code. And `int` indicates that the function will return an integer.

```
1
2  #include <iostream>
3  using namespace std;
4
5  int main(){
6      cout << "Hello World!\n";
7      return 0;
8  }
9
```

Daily Code Jumpstart Choreography

	Mon	Tues	Wed	Thurs	Fri
9am-10am	---	Coaching aims	Daily Aims and Objectives	Daily Aims and Objectives	---
10am-13oo	---	Self-study Time	Self-study Time	Self-study Time	---
Break	---	Social Lunch	Social Lunch	Social Lunch	---
14oo-17oo	---	Self-study Time	Self-study Time	Self-study Time	---
16oo-18oo	---	QandA with Coach	QandA with Coach	QandA with Coach	---

Day 1 Activities

- Activity 1 - Install and setup environment (approx. 45 mins)
- Activity 2 - Get familiar with the syntax (approx. 15 mins)
- Activity 3 - Getting user's input from console and build a calculator (approx. 60 mins)

- Codes for today: Github

Day 1 - Activity 1

Install and Setup Environment

You will need an IDE (integrated development environment) to run and debug C++ applications.

An IDE is not just a text editor, it have the programming environment needed for compiling a program.

In the first activity, download and set up an IDE, create a "hello world" program. ->> next page

Day 1 - Activity 1

Install and Setup Environment

If you're on MacOS:

Getting Started with Xcode (Follow **Way 1**)

If you're on Windows:

Getting Started with Visual Studio (Follow the chapter **"Setting Up Visual Studio"**)

If you're on Linux:

You already have everything you need, just make sure your Linux distribution has the latest GCC.

Writing and Compiling C++ on Linux

Day 1 - Activity 1

Hello World

Create a "hello world" program (approx. 15 mins)

If you're on MacOS, follow this tutorial:

Set up a program in Xcode

If you're on Windows, follow this tutorial

Set up a program in Visual Studio.

Day 1 - Activity 2

Get Familiar with the Syntax

Basic Building Blocks (approx. 10 mins)

You might already be familiar with **data types** (int, float...) / **operators** (+, -, &&, ==…) in C++ since they have lots of similarities with JS and Python.

However, the syntax might be slightly different.

Have a quick look at Chapter 2 Using Data in C++

If you're looking for a tutorial, W3School is a great place to go.

Day 1 - Activity 3

Getting User's Input from Console

Task 01 - Getting User's Input (approx. 15 mins)

Watch and follow this tutorial from 59:41 to 1:09:31

Task 02 - Build a Calculator (approx. 10 mins)

Watch and follow this tutorial from 1:55:58 to 2:02:20

```
Enter first number: 1
Enter second number: 3
1 + 3 = 4
```

Day 1 - Activity 3

Getting User's Input from Console

Task 03 - A Better Calculator (Optional)

Can we extend the calculator to more operators? (e.g., +, -, /, *)

Can we continuously take user's input, and apply operations on the previous result?

Can we add an 'AC' (all clear) command like the one we see on a regular calculator, which clear the previous result and start a new session when the user types 'ac' as an operator.

Hint: a `while (true)` loop might be a good choice, also think about the decision tree in your program

```
First number:   1
Operation:      +
Second number:  3
1+3=4

Operation:      -
Second number:  2
4-2=2

Operation:      *
Second number:  10
2*10=20

Operation:      ac
Cleared

First number:
```

Day 1 Resources

Codes for today

C++ documentation https://www.cplusplus.com

C++ tutorials on W3School https://www.w3schools.com/cpp/default.asp

★★★ C++ Cheat Sheet ★★★

https://github.com/mortennobel/cpp-cheatsheet

Further reading: Statically v. dynamically v. strongly v. weakly typed languages (optional)
https://www.educative.io/answers/statically-v-dynamically-v-strongly-v-weakly-typed-languages

Further reading: compiler and interpreter (optional)
https://www.geeksforgeeks.org/difference-between-compiled-and-interpreted-language/

Day 1 Deliverables

Have an IDE set up on your machine and make sure your Hello World program runs

Familiar with the routine of creating a C++ program

Familiar with the basic syntax in C++

Day 1 de-brief

- How was today for you?
- What has gone well?
- What went as planned?
- What surprised you?
- Did you find today difficult?

- Share anything you made?
- Ask around the class and see if they have anything to share?

ual: creative computing institute

Day 1 Survey

https://artslondon.padlet.org/hbrueggemann/j2yr3zfwkap4v4rq

The password is **Jumpstart**.

Thank you, see you tomorrow at TIME