

Session #1

Introduction

ICPC SCU

Session Structure

- What is programming?
- What does it actually mean to solve a problem?
- How do we communicate with the computer? How to use the stupid machine to assist with computation heavy problems?
- Low/high level programming languages, does it actually matter?
- Why C++?
- Building blocks of a program (how do languages make your life easier?)
- Headers and Includes
- Compilation process
- What is an algorithm?
- How will we describe the steps of our programs/solutions

What is programming?

- It refers to the technological process for telling a computer which tasks to perform in order to solve problems
- You can think of it as a collaboration between humans and computers, in which humans create instructions for a computer to follow (code) in a language computers can understand
- Fundamental rules of programming are simple and clear, but programs built on top of these rules tend to become complex enough to introduce their own rules and complexity
- A program is many things, but we can define it as a piece of code typed by a programmer, and it is the directing force that makes the computer do what it does
- It is data in the computer's memory, yet it controls the actions performed on this same memory.

What does it actually mean to solve a problem?

- To solve a problem in programming means to create a program that can perform a specific task or achieve a desired outcome, given some inputs and constraints
- Solving a problem may involve:
 - Identifying the problem: understanding the problem, its input and output, requirements and expectations
 - Devise a plan: come up with a strategy or an algorithm to solve the problem, using logic, data structures, and programming concepts
 - Carry out the plan: implement the plan in code, using a programming language and tools of your choice. Test and debug the code to ensure it works correctly.
 - You may need to look back for evaluation, refactoring, optimizing, maintaining

How do we communicate with the computer? How to use the stupid machine to assist with computation heavy problems?

- To communicate with the stupid machine, you need to use a language it understands.
- You have to use a language that you can understand as well
- What if there is something that works to compile a language you understand to a language the machine understands?
- A programming language is a set of rules and symbols that instruct the computer to perform certain tasks

Low/High level programming languages, does it actually matter?

- Writing low-level/machine code is not a trivial task ...

```
- 48 65 6c 6c 6f 2c 20 77 6f 72 6c 64 21 0a
- 48 c7 c0 01 00 00 00
- 48 c7 c7 01 00 00 00
- 48 c7 c6 00 10 40 00
- 48 c7 c2 0e 00 00 00
- 0f 05
- 48 c7 c0 3c 00 00 00
- 48 c7 c7 e8 ff ff ff
- 0f 05
-
```

- What about looking at a high-level one?

```
print("Hello, world!")
```

Why C++?

- Fast and efficient: can execute complex algorithms with less time and memory consumption
- Offers a library called STL (Standard Template Library)
- C++ supports multiple paradigms: object-oriented, imperative, functional, ..
- Has a large community: provides help, tips, tricks, resources

Building blocks of a program

- Data and ways to deal with data
- Simple instructions for doing repetitive tasks to let us reduce the amount of thinking about the implementation
- Control structures, to control the flow of your programs
- Subroutines: blocks of code that can be defined once and reused multiple times in a program

Headers and Includes

- Headers are files that contain declarations or blocks of codes and can be used by different source files
- In C++, headers usually have the extension .h and included with the preprocessor directive `#include`
- Includes are the act of inserting the contents of a header file into a source file using the `#include` directive using `<>` or `“”`
- The purpose of headers and includes is to separate the interface from the implementation, to avoid code duplication, to improve readability

What is the compilation process?

- The compilation process is the process of converting the source code of a programming language into an executable file that can be run on a computer
- Consists of several phases includes intermediate code generation, optimization, and machine code generation
- Some compilers may perform the operation differently from the others, but the process in general is the same

What is an Algorithm?

- An algorithm is a set of rules or steps that can be followed to solve a problem or perform a task
- An algorithm can be written in natural language such as English
- The algorithm can be simple or complex depending on the nature of the problem and the solution
- You write an algorithm for almost any process
- Can you come up with an algorithm for finding the maximum number out of a set of numbers?

How will we describe the steps of our future solutions?

A tweak of writing actual algorithms