Agenda - Introduction to programming in C++

Lyon College, Spring 2023

Marcus Birkenkrahe

February 24, 2023

FANTASTIC! WHAT YOU ALREADY LEARNT IN ONLY 6 WEEKS!

TERM	MEANING
Meta data	Data about data (e.g. layout instructions)
Infrastructure	Computing roadworks: hardware and software
Editor	Program to write programs in
Compiler	Program to turn source into machine code
`gcc`	The GNU C compiler
Source code	Code for humans to read and edit (`.c`)
Machine code	Code for machines to execute (`.exe`)
Git	Software version control system (2005)
Emacs	Extensible editor written in Lisp (1985)
FOSS	Free and Open Source Software
Linux	FOSS operating system (1991)
Windows, MacOS	Commercial OS (Microsoft, Apple)
Android	Linux for smartphones (Google)
iOS	MacOS for Apple smartphones
Command line	Terminal, shell program to talk to the OS
Prompt	Location on your computer, e.g. `C:\User\`
Raw file	No control characters for syntax highlighting
Syntax highlighting	Making programming language visible
`DIR`	Windows command to list files
`cd`	Command to change directory
Literate pgm	Doc + code + output for humans and machines
Org-mode	Plugin for Emacs to edit Org files (`.org`)
Dunning-Kruger effect	Illustrating ignorance of your own ignorance
DEC PDP-11	1970s mainframe computer
UNIX	Operating system (ca. 1969)
ANSI	American National Standard Institute
Assembler	Machine code (hard to write/read)
Algorithm	Fixed process or set of rules
Linux	Operating system (ca. 1991)
C	Imperative, procedural programming language
C++	Object-oriented (OO) superset of C
Clang	C/C++ compiler
gcc	GNU compiler bundle (incl. C/C++)
Java,C#	OO programming language
Perl	Scripting language
Git	Software version control system
GitHub	Developer's platform (owned by Microsoft)
Library	Bundle of useful functions and routines
Portability	Ability of software to run on different hardwares
Efficiency	Software speed of execution and memory requirements
Permissiveness	Degree to which a language tolerates ambiguities
Object-orientation	Ability to define abstractions
System programming	Programming close to the machine
Application programming	Programming close to the user

TERM	MEANING
Linker	translates object code to machine code
Syntax	language rules
Debugger	checks syntax
Directive	starts with `#`, one line only, no delimiter
Preprocessor	processes directives
Statement	command to be executed, e.g. `return`
Delimiter	ends a statement (in C: semicolon - `;`)
Function	a rule to compute something with arguments
String	Sequence of *character* values like `hello`
String literal	Unchangeable, like the numbe `8` or the string `hello`
Constant	Set value that is not changed
Variable	A named memory placeholder for a value, e.g. `int i`
Data type	A memory storage instruction like `int` for integer
Comment	Region of code that is not executed
Format specifier	Formatting symbol like `%d%` or `%f%`
Data type declaration	Combination of type and variable name - e.g. `int height;`
`int`	C type for integer numbers, e.g. 2
`float`	C type for floating point numbers, e.g. 3.14
`char`	C type for characters, like "joey"
Formatting	Tells the computer how to print, e.g. `%d` for `int` types
`%d`	Format for integers
`%f` and `%.pf`	Format for floating point numbers (with `p` digits after the point)
`#define`	Define a constant with the preprocessor, e.g. `#define PI 3.14`
`math.h`	Math library, contains mathematical constants & functions
`stdio.h`	Input/Output library, enables `printf` and `scanf`
`const`	Constant identifier, e.g. `const double PI = 3.14;`