1) Define the following terms:

(a) Computer Program:

(b) Algorithm:

(c) Documentation (as it relates to a computer program):

(d) Machine Language:

(e) Assembler:

(f) Compiler:

2) Describe the difference between the Problem Solving Phase and the Implementation Phase. Explain why it is important not to be tempted to bypass the Problem Solving Phase and go straight to the Implementation Phase.

3) Scan the section 1.3 on What's Inside a Computer. There are no questions for this topic, but I want you to get a cursory understanding of the main parts. This will be the central topic in CS281, but it is good to have a basic understanding for CS173.

4) What is the difference between how a C/C++ program is run versus a python program?

5) What is software maintenance?

6) Define Software Engineering.

7) Why is software engineering an important aspect of computer programming? What can go wrong if we don't embrace the principles of software engineering?

8) Describe the Divide-and-Conquer approach to problem solving.

9) Who invented the C programming language and when? Who invented the C++ programming language and when?

1) Define syntax and semantics.

2) Explain why there must always be a main() function in every C/C++ program. Explain the purpose of the return 0 statement at the end of main().

3) What are the rules for valid C identifiers?

4) What is a C variable declaration? What purpose does it serve? Why don't we have type declarations for variables in python?

5) Give an example or two of some data that might be stored in each datatype:

(a) int

(b) float

(c) char

(d) string

6) What does the C++ preprocessor do?

7) How do we signify "special characters" for output? Give three examples of special characters.

8) What is the difference between a reserved word and a constant in C++?

9) What is the purpose of the `#include <iostream>` statement?

10) What is the purpose of a namespace?

11) Convert the following binary values to decimal representation:

(a) 10101_2 (b) 10101101_2

12) Convert the following decimal numbers to binary representation:

(a) 777_{10} (b) 101_{10}

List the linux commands below which perform the indicated task:

(a) List the contents of the current working directory.

(B) Print the current working directory path.

(C) Change back one directory (to the parent of the current directory).

(D) Create a directory named cs173.

Explain why every c++ program must have a function named main().

What is the purpose of the following statement in c++? Be specific about the purpose of both #include and <iostream>.

Convert the following decimal number to binary and binary number to decimal.

(a) 567_{10} (b) 10111_2

Linux Commands

pwd – print working directory, tells us where we are
ls – lists everything in the working directory (or a specified directory)
cd [path to a directory] – change directory, move where we are
mkdir [path to a directory] – make a new directory
touch [path + filename] – makes a new file
python [path + filename] - runs a .py file
g++ [path + filename] -- compiles a .cpp file

- Use -o [ofilename] to specify an output file name
- If no output file is specified, a.out is the produced file

./[ofilename]

- Runs an executable file

mv [source_path + filename] [destination_path + filename] -- moves a file (possibly renames it)
rm [path + filename] -- removes a file

include <iostream>

- Preprocessor directive, similar to import
- C standard library split into header files
- iostream header file lets us use input and output
- Contains istream and ostream objects
    - Ostream object: cout
        - Insertion operator: << puts output to an ostream
        - endl manipulator: moves to the next line (inserting a newline character

Don't need def keyword to define functions
Before the name of a function, we need to specify the type it returns

- int
- void (doesn't return anything)
- float/doubles
- char
- bool
- string

Syntax differences:

- Instead of whitespace and colons, use {} to denote code blocks
- Statements end with ;
- Comments denoted with //

main() functions in C++ are required for a program to run, entry point for the operating system
- must return an int

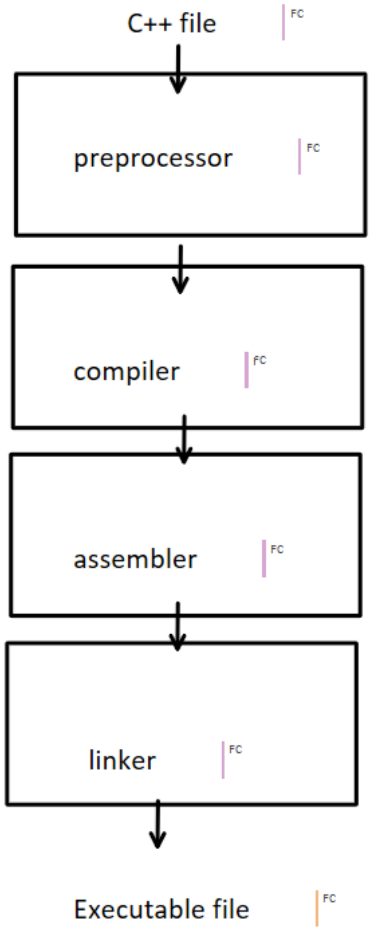
- Returning 0 indicates success
- Get called when the program runs

Python is compiled in little chunks and run on a virtual machine, interpreted so it runs line by line until an issue is found

- Portable

C++ gets compiled/translated into machine language, produces an output file which can be run on a specific machine

- Compiler can help us find a lot of issues
- Faster!
- Output file produced on one machine can't be run on other machines



More detailed breakdown of the steps of the compilation process

|

V

Handles preprocessor directives, copies over content from included header files

Translates into assembly language
- More in 281!

Translates into machine code (.o files)
- More in 281!
- But we'll talk a bit about machine code
- Binary

Combines multiple .o files

1 byte = 8 bits

Range of values that can be stored in one byte:
00000000_2 to 11111111_2

Values that can be stored in *n* bits:
$0 \ldots 2^n - 1$

Types in C++ are represented with a specific number of bytes:
chars (1 byte)
ints (often 4 bytes) vs. shorts (often 2 bytes) vs. longs (at least 4 bytes, can be more)

Types can be signed (default for numerical types) or unsigned:
To represent signed types:
If leftmost bit is 1--value is negative (-2^{n-1} to -1)
If leftmost bit is 0--value is nonnegative(0 to 2^{n-1} -1)