

# CS 1301 Study Guide

-or-

What you should know (a non-exhaustive list):

## Computer Science:

What is a program? An algorithm? How is Computer Science (CS) different from Math? How is it similar? Why is it important to understanding how computer programs are made? What makes a computer program correct? How do we test if a program is correct? What are the different ways a program can “misbehave”?

How long have computers been around? The idea of computation? What were early computers used for? How have computers changed over time?

You should know all of the computer science vocabulary introduced in both of your textbooks.

What is recursion? What is needed for a correctly working recursive function?

What is iteration? Why might you want to iterate in a computer program, and over what?

What is conditional execution? What does conditional execution allow us to do?

What does it mean to call a function or method? What does it mean to “pass” a parameter into a function? What does it mean to return a value? What is the difference between returning a value and printing output? What is the flow of execution? How is it affected by iteration, conditionals, function calls and returns? What is the main idea behind “Functional Programming”? What is the main idea behind “Object-Oriented Programming”? How are they different from “Imperative” or “Procedural” programming?

How is data represented inside modern computers? How do you convert between decimal, octal, hexadecimal, and binary representations of integers? What are different data “types”?

What does operator precedence mean? How does it affect how expressions are evaluated?

How do boolean expressions work? What are they used for? Give an example of short-circuit evaluation. How do the AND, OR, and NOT logical constructs behave? Draw a table showing all possible inputs and corresponding outputs for AND, OR, and NOT.

How is data stored on non-volatile memory (Hard Disk, Flashdrive)? Why might you want to store data on a non-volatile system instead of just keeping it all in RAM? (What is RAM, how is it different from a Hard Drive or Flashdrive?) What is a file-system?

What is computational complexity (Big O notation)? What is it good for? Why do we study computational complexity of different algorithms? What are the computational complexity classes that we learned about in class? What is an example algorithm in each complexity class? Draw a size vs work graph for the computational complexity classes we learned about. Is searching always faster than sorting? If it is, why does anybody ever sort data? How do the Bubble, Insertion and Merge sorting algorithms work? How does a Binary Search work, and how is it different from a Linear Search?

## Python:

What is Python? Where did it get its name? What does it mean for a programming language to be Dynamic?

What are four basic data types that we learned about in Python? What are four compound data types we learned about in Python? (What does it mean to be a compound data type?) Which data types are mutable and which are immutable? What does that mean in practice? How is a Dictionary different from a Tuple or a List? How is a Tuple different from a List? When is a Tuple useful?

What is a valid name for an identifier? How does assignment work? What does it make an identifier do? What happens when you assign something to an identifier that is already pointing at something else? What happens when you execute a statement like the following: `myName = myName + 1`? What is an alias? How do you create an alias? How do you create a copy (for example, of a list) instead of an alias? Why does it matter? Give an example where using an alias produces very different results from using a copy.

How do you define a function in Python? How are parameters defined? How do you return a value? How does the return statement affect the flow of execution? How do you print messages and data to the console? How do you create a string with inserted data using the format specifier? How can you make an identifier point at a function? (Why would you want to?) How do the MAP, FILTER, REDUCE operations work? How do you create a temporary, unnamed function?

How do conditional statements work in Python? (IF, IF/Else, IF/ELIF/ELIF/ELSE....) What are two different ways to iterate in Python? Can you define a recursive function in python? How do these constructs affect the flow of execution? What is an exception, and how do you handle one?

How do the bracket operators work? (e.g. `myList[5]` does what?) What happens when you use a negative number as an index in a bracket? What is a slice, and how do you take one? What default values can you use to indicate the beginning or end of a compound data type when using the slice operator? What is a shortcut to take a complete slice (full copy)?

How do you open a file? Read from a file? Write to a file? How do you know when you reach the end of a file you are reading?

## The Robot & Myro:

What actuators (output devices) does your robot have? What sensors does your robot have? Which are located on the Scribbler, and which are located on the Fluke? What are the API calls to access the sensor values or activate the actuators?

What is Myro? How is Myro related to your robot? What other features does Myro offer that have nothing to do with your robot?

How are pictures represented? How do you access and modify pixels in a picture? How do you copy a picture? What is different between working with Pictures and working with the Graphics Object Interface? How are graphics objects represented that is different from pictures?