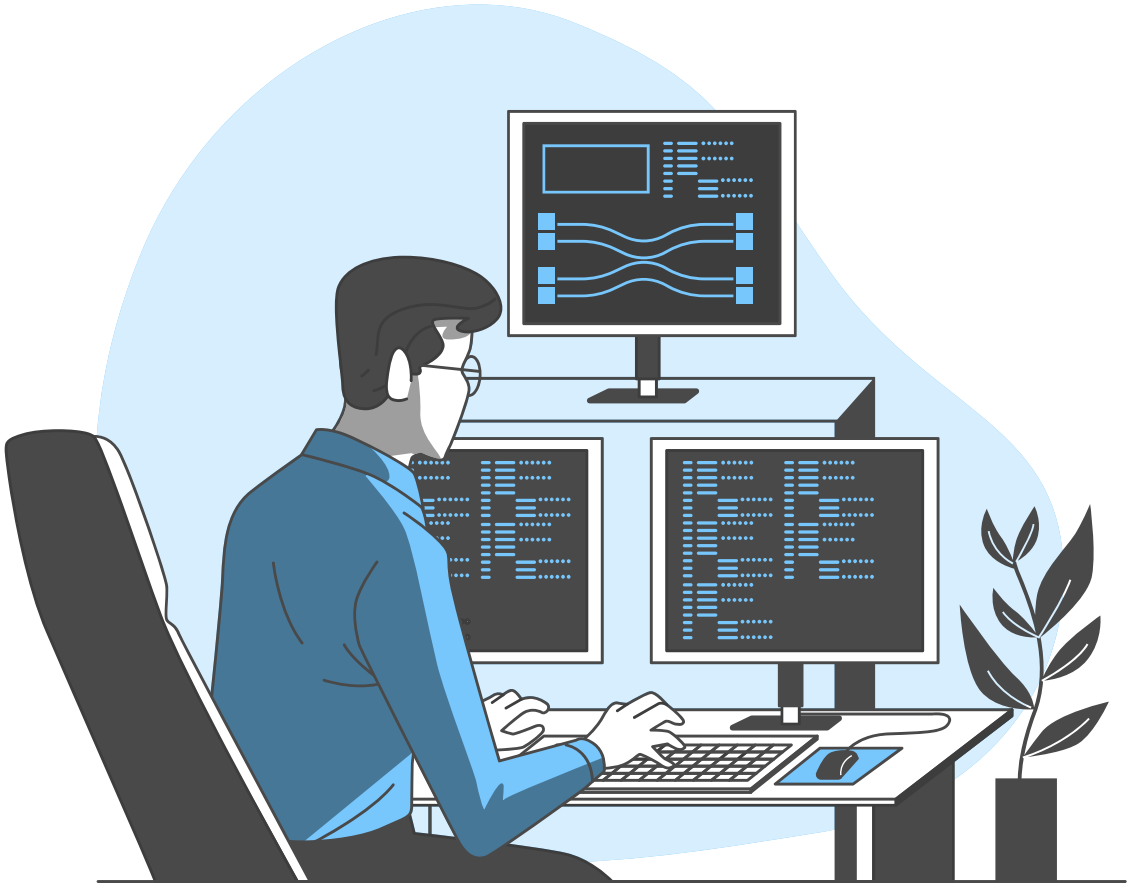


From the Bottom Up Series Introduction

01

What are your preferred learning methods?

For example, how did you learn to swim?

Did you get thrown into the deep end?
Maybe you did, and maybe that was okay. But odds are you started out from the basics and moved up.
That's how we're going to go about learning how a computer functions.

...

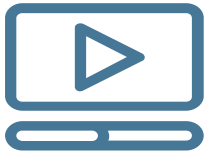
How do computers work?

This is a huge question!

This question is huge and it can be split into two parts:

1. How does the hardware of a computer work?
2. How does the software of a computer work, and how does it work in conjunction with the hardware?

The content of the answers to these question is enough to fill an entire course.. Actually it is enough to fill many many courses. And it does.



In this section of the class..

We'll be hitting the very basics of every step of how a computer works, from the smallest bit up to the operating system.

I know it seems like way too much at first, but the goal is just to give you an introductory idea of how that box in front of you is working.

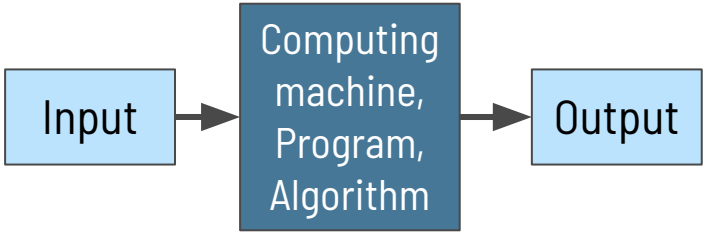
Where you go from there is up to you!

... ...

If nothing else, your computer vocabulary will be strengthened and you will have a better understanding of your machine.

Where are we going in the next 7 lectures?

The Theory of Computation



01

At this point, we don't know the specific of what is going on in that middle box. We don't know how it is constructed. But we don't need to!

This computing box is general enough to describe any kind of computation, from searching for a name in a directory to factoring a 300 digit number.

However, some computational problems are harder to solve than others in that they take a lot more time or a lot more resources to solve. What makes a problem harder to solve than another?

There is also the notion of general computability: is it even possible to solve this problem with a computer?

How can we represent this complex computer with a simple machine?

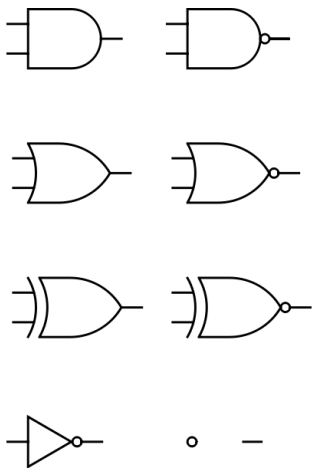
These are the questions that the theory of computation explores. Again, it is enough to build a whole course on, but we are simply introducing it!

Binary, Boolean Logic, Boolean Arithmetic

02

0 1

False True



These are the smallest building blocks of the computer! Strings of 0's and 1's represent data on the computer and also control the functions of the computer.

In this lecture we'll learn how to read binary, and how it is used to represent numbers and other things.

Many forms of arithmetic can be used to manipulate binary numbers.

Binary numbers (0's and 1's) are the inputs to and outputs of these little plug looking things called gates.

Gates are chained together to make more complex devices that eventually make up the processor and something called the "arithmetic logic unit" in a computer.

Booleans to memory

03

Now we come to an idea that I'm sure you all have heard of: memory. Whether you've heard of it in the context of your phone where the memory is always running out, or even your own brain and how it stores your memories and thoughts.

The logic gates that we discussed in the last lecture can be arranged to represent a device that stores bits (0's and 1's). The most basic idea is a **register** that stores a single bit (0 or 1). Then memory can scale up to registers which hold a larger amount of bits, and then to something called RAM.

In this lecture we will talk about memory and why it is important: It holds the data and the instructions for the computer that we care about. It is also one of the major things that goes into the makeup of a computer.

Computer Architecture & Program Control

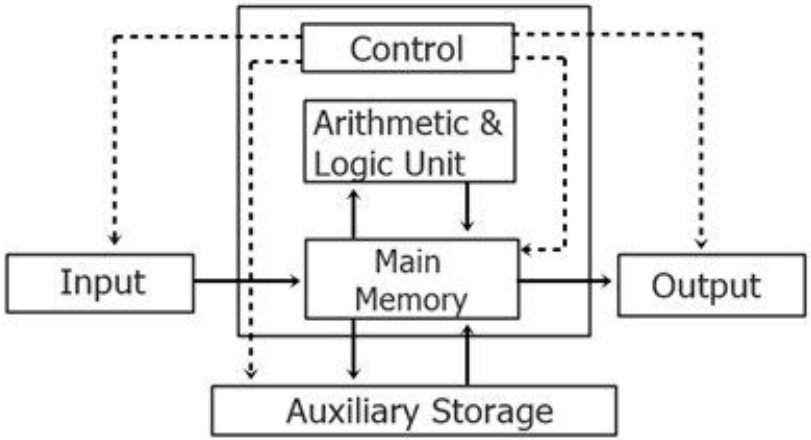
04

The boolean logic gates and other components that are built out of them all work together to create the CPU.

The CPU runs on a cycle, constantly fetching things called instructions from the memory, decoding the instructions so it knows what to do, and executing the instructions.

It continues on this cycle over and over and over again.

The CPU consists of the unit that controls the fetch/execute cycle, the arithmetic and logic unit, and the main memory.



Machine Language

05

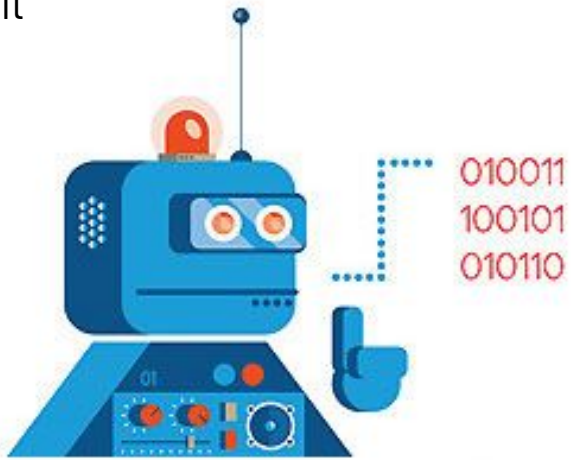
Somehow we're coming back to those 0's and 1's again!

Remember when we said there was a difference between "instruction" memory and "data" memory. Well, those "instructions" are still made up of 0's and 1's, and this is called "machine code" .

Machine code is the input to the processor that tells it what to do.

The processor interprets machine code as instructions for:

1. Moving data around
2. Performing arithmetic on data
3. Performing logical operations on data (remember gates?)
4. Controlling the program flow



Assembly Language

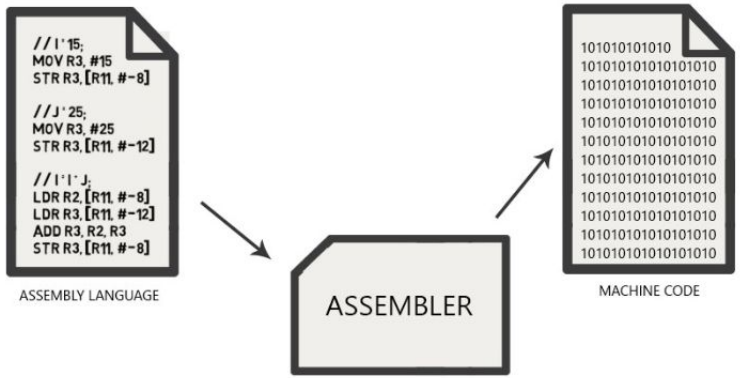
06

After we learn how the 0's and 1's correspond to instructions for the processor, we can move a step up into a representation of the instructions that is more human friendly.

Remember, machine language is at a lower level than assembly language, so from our point of view, assembly is translated down into machine language.

There are different types of assembly languages but they all have the same basic concepts.

The instructions, just like machine language, move data in and out of registers, perform arithmetic operations on data, and control the flow of the program.



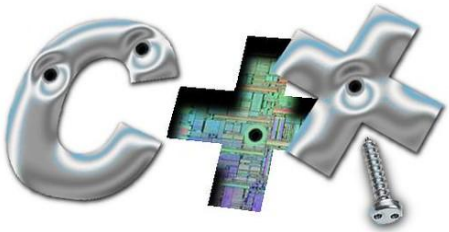
High Level Languages

07

After we get through the hardware, the machine language, and then the most understandable form of instructions so far, assembly language, we come to what you probably think of when you think about “coding”.

High level languages! These languages translate directly down into assembly languages. They all have some very similar qualities and they can also be very different.

We’ll talk through some examples of high level languages and then we’ll see how these languages are used to solve problems and make things!



That's the big picture!

There is no way possible for us to cover every single detail of all of those topics in an introductory course. The purpose of this series is to give you a very high level overview of how your computer functions. As discussed previously, there are courses down the road that go into great detail in all of these topics, but for some of you this is all you need to know!