An Introduction to Computers

Overview

- Logic
- Bits and Bytes
- Algorithms

A brief introduction to Logic

- Logic is a science that sets forth rules for properly-ordered thinking
  - Helps with identifying faulty reasoning (fallacies)
  - Helps with verifying a claim's validity

    The only true way to verify is to test!
- Critical Thinking is the application of those rules
  - Based on criteria that have been testing and verified
  - Logically combines those criteria to form an argument

Why talk about Logic

- Logic can be represented in any number of ways
  - Through words
  - Through math
  - Through code
- Computers fundamentally work through logic

How?

The Turing Machine

- One of the simplest ways of representing logic is through a simple switch.
  - *On* or *Off*. *Yes* or *No*. *True* or *False*. *1* or *0*.
- A Turing machine consists of a large number of these switches.
- For instance, let's think of a light that can have 4 states:
  - Off
  - Red Light
  - Yellow Light
  - Green Light
- How many switches would we need to use to produce these states?

Switches

- Two. Merely two. How can we do that?
- Let's call our switches Left and Right. Each can be Up or Down.
- So, here is an example of how we could match each state:
  - Off – Left Down, Right Down
  - Red Light – Left Down, Right Up
  - Blue Light – Left Up, Right Down
  - Yellow Light – Left Up, Right Up

A bit less wordy

- Okay, great, we were able to set up that light!
- That being said, out solution was a bit… Wordy.
- So, rather than saying Left or Right, let's just say Up or Down for each
- For Example:
  - Off – Down, Down
  - Red Light – Down, Up
  - Blue Light – Up, Down
  - Yellow Light – Up, Up

Can we go further?

- Alright, looking good!
- That being said, let's see if we make that *even shorter*
- Let's use 1 for Up and 0 for Down.
- Also, let's get rid of the comma.
- For Example:
  - Off – 00
  - Red – 01
  - Blue – 10
  - Yellow – 11

Even more complex

- Congratulations! What you just saw is Binary.
- That being said, what if we wanted to combine colors?
- For instance, combine Red and Yellow to get orange.
- In total we have these states:
  - Off
  - Red
  - Purple
  - Blue
  - Green
  - Yellow
  - Orange
  - Black
- How many switches would that take?

More Switches

- One answer would be 3.
- In other words each switch represents a color.
- If a switch is off, that color is not being shown.
- For Example:
  - Off – 000
  - Red – 001
  - Blue – 010
  - Purple – 011
  - Yellow – 100
  - Orange - 101
  - Green - 110
  - Black - 111

Bits and Bytes

- In Computer Science, a Switch (0 or 1) is called a Bit
- So, our previous color scheme was represented by 3 Bits.
- A Byte, on the other hand, is represented by 8 bits.
  - For a total of 2^8 or 256 possible combinations
  - This is the true building block for computers
- Representing letters:
  - A – 01000001
  - B – 01000010
  - C – 01000011

Make it less wordy!

- As you might imagine, having to type out 01000001 for A is very tedious. As a result of this, Computer Scientists came up with a different way of representing Bytes.
- As it turns out, one Byte can be represented by two Hexadecimal Characters. Hexadecimal goes from 0-9 and then from A-F.
- So, for instance our capital letters from before would become:
  - A – 41
  - B – 42
  - C – 43

Takeaways for Binary and Hex

- At their core, computers operate through many switches
- In order to represent their values more easily, we use Hexadecimal
- Through logic, combinations of these switches can be bound to different functions (storing numbers, displaying text, etc)
- Higher level programming languages are compiled—turned into hexadecimal—in order for programmers to communicate with machines through logic.

Algorithms

- If hexadecimal is how computers “think” and programming languages are how we communicate with them, Algorithms are what we tell them to do.
- An algorithm is a repeatable process, or in other terms, a set of steps.
- This process is meant to take certain inputs and consistently produce a set of outputs.
- For example:
  - Printing out a PDF document from your laptop
  - Writing an email

Steps of an Algorithm

- Printing out a PDF Document:
  - Inputs – A PDF Document, a Printer
  - Output – A paper Document
  - Step 1 – Open the Document
  - Step 2 – Press Ctrl+P
  - Step 3 – Select a Printer
  - Step 4 – Click the “Print” button

Don't forget this

- Remember: computers, by nature, operate through a giant series of switches. That means they can't come up with anything on their own.
- In other words, you need to define EVERTHING
- For our previous example:
  - What is a PDF Document, how can we read it, and where is it?
  - What do we use to open it?
  - What does Ctrl+P mean?
  - What is a Printer and which one are we supposed to select?
  - How do we physically print it?

Don't despair

- Thanks to object-oriented programming, we can use and refer to things that other people have already made.
- For instance:
  - Rather than making your own file-reading code, use java.io.FileReader!
  - Rather than making code to communicate with a printer, use javax.print!
- And don't forget. This applies to code you've already made.
- So if you've done something before, use it to do something greater.

Summery Review

- Logic
  - What is logic?
  - How it works?
  - Binary logic
- Bits and Bytes
  - Bit
  - Byte
  - Hexadecimal
- Algorithms
  - Set of steps

Questions?