Introduction: How Computer Stores Information (0 – 2:30)

- Everything in computer is encoded as 0s and 1s (bits)
- 0s and 1s are stored in hard drive
- How does hard drive work?
    - Inside hard drive are disks called platters
    - Platters are coated with tiny magnetic particles
    - When particles are aligned one direction (e.g. North-South), interpreted as a 1
    - When particles are aligned other direction (South-North), interpreted as a 0
- Hard drives store information in the same way as floppy disks…

Video: How Computer Reads Floppy Disk (2:30 – 4:00)

- First, floppy is inserted into floppy drive
- Levers inside computer push back metal shutter to reveal "cookie"
- Cookie is a film disk coated with magnetic particles
- Levers containing read/write heads pinch cookie so that they are almost touching
- One motor rotates cookie based on commands from PC
- Second motor moves read/write heads across cookie
- First checks write protect tab in corner of disk using light to determine if open or not

Demo: Open Floppy Disk (4:00 – 6:00)

- Slide shutter to reveal cookie
- Pull off shutter
- Take off plastic cover
- Inside, just a film disk
- Take away: look how simple!

Video: How Is Data Written to and Read from a Floppy? (6:00 – 7:30)

- Computer sends pulses of electricity to head
- Heads act like electromagnets
- Create electric fields to reach surface of disk
- Electric fields alter coding of magnetic particles in coating of cookie, and so data is written
- Direction of current determines polarity (1 or 0)

Counting in Binary (7:30 – 16:00)

- So hard drives give us the ability to store 0s and 1s. This is useful only if we can translate information into 0s and 1s.
- First, we need to be able to represent numbers using only 0s and 1s. We do this by putting them into binary.
- Suppose we have eight light bulbs (or transistors) that can be in the on or off position to represent a 0 or 1.

- How do we represent 0?

  OFF OFF OFF OFF OFF OFF OFF OFF

- How do we represent 1?

  OFF OFF OFF OFF OFF OFF OFF ON

- How do we represent 2?

  OFF OFF OFF OFF OFF OFF ON OFF

- How do we represent 3?

  OFF OFF OFF OFF OFF OFF ON ON

- Light bulb representation can be read as binary numbers.

  0:
  OFF OFF OFF OFF OFF OFF OFF OFF
  0 0 0 0 0 0 0 0

  1:
  OFF OFF OFF OFF OFF OFF OFF ON
  0 0 0 0 0 0 0 1

  2:
  OFF OFF OFF OFF OFF OFF ON OFF
  0 0 0 0 0 0 1 0

  3:
  OFF OFF OFF OFF OFF OFF ON ON
  0 0 0 0 0 0 1 1

- Notice that when we use two light bulbs (or bits) we can only go up to "3". To represent "4", we must use an additional light bulb (bit)

  OFF OFF OFF OFF OFF ON OFF OFF
  0 0 0 0 0 1 0 0

- To figure out what a binary number is in decimal, think of columns as 128s, 64s, 32s, 16s, 8s, 4s, 2s, and 1s in the same way we normally have 1000s, 100s, 10s, 1s:

  128s 64s 32s 16s 8s 4s 2s 1s
  0 0 0 0 0 1 0 0

- Based on the above, we can calculate 100 = 1*4 + 0*2 + 0*1 = 4, so 100 in binary translates to the number we know as 4 in decimal.
- Exercise: Translate the numbers 0 2 1 3 8 into binary
- Answer:
  00000000
  00000010
  00000001
  00000011
  00001000
- We can also use binary to encode letters using ASCII. Corresponding to each lower case letter, upper case letter, symbol, and number is a decimal number between 0 and 255.
- For instance, A is 65 and a is 97.
- We can take each number's decimal number and translate it into binary. This binary number is the ASCII representation of the number.
- For instance, A is 0 1 0 0 0 0 0 1 and a is 0 1 1 0 0 0 0 1.

Course Expectations, Requirements, Procedures etc. (16:00 – 33:00)

Course Overview (33:00 – 49:00)

- Course will focus on C. Many programs on your computer are in C, C++, or C#.
- Week 0: First, to avoid distraction of syntax, will introduce graphical programming environment called Scratch. For those who have never programmed before, you can create some dynamic or interactive program in just a week's time. (Which you can't ordinarily do in C or Java.)
- Week 1: Introduce C, more traditional, arcane language.
- Week 2: Explore C at more sophisticated level. Use cryptography as context to learn functions and other concepts of programming.
- Week 3: Canonical problems of searching and sorting
- Week 4: How to design software well and debug software
- Week 5: Look at how program runs in more technical detail: memory, RAM. Look at it in the context of digital forensics.
- Week 6: Data structures.
- Week 7: Data compression. How do you take very big file and compress it down to something much smaller?
- Week 8: What goes on in CPU? ANT – assembly language. Lowest level we can get before 0s and 1s.
- Week 9: Secure coding.
- Week 10: Introduction to web-based programming and PHP. SQL for database queries. Will be able to make dynamic webpages.