Running times of algorithms:

Suppose algorithm is $O(n)$, takes 3 seconds for $n = 100$.

- # sec for $n=200$? $\rightarrow$ $3 \times 2 = 6$ sec
- # sec for $n=700$? $\rightarrow$ $3 \times 7 = 21$ sec
- # sec for $n=1000$? $\rightarrow$ $3 \times 10 = 30$ sec

Suppose algorithm is $O(n^2)$, takes 5 seconds for $n = 2000$.

- # sec for $n=4000$? $\rightarrow$ $5 \times (2^2) = 20$ sec
- # sec for $n=14000$? $\rightarrow$ $5 \times (7^2) = 5 \times 49 = 245$ sec
- # sec for $n=20000$? $\rightarrow$ $5 \times (10^2) = 500$ sec

Challenge: Suppose algorithm is $O(2^n)$, takes 5 seconds for $n = 10$.

- # sec for $n=11$? $\rightarrow$ $5 \times 2^1 = 10$ sec
- # sec for $n=15$? $\rightarrow$ $5 \times 2^5 = 160$ sec
- # sec for $n=20$? $\rightarrow$ $5 \times 2^{10} = 5120$ sec

Topics to explore:

Programming Languages

- Java - Used for web applications. Object-oriented. (Used in CS 201)
- C - Used for systems programming. Low-level, imperative. (Used in CS 202)
- Javascript - Used to embed code in web pages.
- Matlab - Used in matrix computations.
- Haskell - Functional language.

Other languages: C++, Ruby, Scheme, Coffeescript, Processing, R

Block-based languages: Scratch, Alice, App Inventor

Programming Environments: Dr Java, Eclipse, Blue J, Pencil Code, command line

Unix

Use of command line interface (google "Unix tutorial")

HTML / CSS

HTML tutorials (google "HTML tutorial", e.g., w3schools.com/html)
CSS tutorials (google "CSS tutorial", e.g., w3schools.com/css)