**Intro to the Internet**

1. How the Internet Works
   1. Data Transmission
      1. What is a protocol?
         1. A ruleset to send/receive certain types of data
      2. Internet Protocol (IP)
         1. Allows computers to locate other computers on the internet
         2. Everything connected to the internet is assigned a unique number (IP address)
         3. Syntax: NNN.NNN.NNN.NNN
            1. Each of the four values ranges from 0 to 256
         4. Localhost: a special IP address (127.0.0.1) always points to your own computer as if another computer was accessing yours from the internet
         5. Websites can have more than one IP address (example: google.com)
      3. Domain Name System (DNS)
         1. The global ‘phonebook’ that turns a text domain into an IP address
      4. Ports
         1. The ‘mailbox’ of the computer, the channel to send/receive data
      5. URL – Uniform Resource Locator
         1. A web address that is useful/rememberable for the user
         2. Syntax: scheme://domain:port/path?query\_string#fragment\_id
         3. Ex: http://ip-address:80/path?query\_string#fragment\_id
2. HTTP (Hyper Text Transfer Protocol)
   1. Text based protocol (human readable)
   2. Most of the internet runs on port 80, or port 443 for secure (HTTPS)
   3. To view a HTTP transmission for a website:
      1. Open terminal, input ‘telnet google.com 80’, then input ‘GET’
      2. GET outputs the homepage of the request when no argument is input
   4. Parts of an HTTP response:
      1. Status – was the request successful? Displays ‘OK’ if yes
         1. Status Codes are messages that are displayed (and assigned by number)
            1. ex: 200 – OK, 404 – Not Found, etc.
            2. Try in terminal ‘curl -i google.com’ vs ‘curl -i www.google.com’
      2. Response Header – Key Value pairs that display information about the response
         1. Cache-Control: information about how the page should be cached
         2. Set-Cookie: a hash to be stored by your browser
         3. Content-Type: defines the response type (i.e. ASCII or Unicode?)
            1. The internet predates Unicode, so at the HTTP level ASCII is common
      3. Response Body – a response written in HTML, what is processed by the browser to display
   5. Making HTTP requests
      1. Use ‘curl’
         1. Syntax: curl -i -X GET google.com
   6. HTTP Methods
      1. GET – list all the things
      2. HEAD – list only the header (no content)
         1. ex: curl -i -X HEAD www.google.com
      3. POST – create a new thing
         1. ex: curl -iX POST --data “” www.google.com
      4. PUT – replace (or update) all the things
         1. ex: curl -iX PUT --data “” www.google.com
      5. DELETE – delete all the things
         1. ex: curl -iX DELETE www.google.com
      6. Many more (TRACE, OPTIONS, CONNECT, PATCH)
   7. REST – Representational State Transfer
      1. The architecture for HTTP REST, defining Methods like GET, PUT, POST, etc.
3. Cookies
   1. Like a sticker that a convention might put on your shirt that has information, like your name or id number
   2. Cookies have an expiration date, when exhausted the cookie is removed/deleted
   3. A shopping cart is an example of a cookie, if your browser didn’t accept cookies the cart would always be empty
   4. HTTP is stateless, meaning that every server request is treated as new – hence why cookies exist
   5. Browsers return cookie info when they submit new requests
   6. Cookies can be used for harm, ex: adds
   7. Cookies can be removed by the user in the browser