CS 151 Introduction to Computer Science Summer 2017 Syllabus and Information

Table of Contents: **table of contents** **Study Guide** **Class Notes**
Instructor Directories: cd ~jkinne/public_html/cs151-su2017/
Course website and information: this google doc | video lectures | lab assignments | HW assignments | w3schools

Feedback on this as an Online Class

- I'll keep some notes here to remember what was liked, not liked, etc.
- Try to keep due dates the same every week. Pick 2 or 3 days to have all due dates, to mimic a traditional course and keep the scheduling simpler
- Time to take online quiz/exam - have a default day/time if people don't ask for a different one.
- Be very responsive to email, etc.
- Talking by skype/phone seems good
- Find a buddy assignment - so people have at least one other student in the class they feel comfortable talking to
- Good to have - videos, chat, reference/book to look at, virtual office hours
- Requirements - make clear up front (before course and at the beginning) that students are required to have a decent internet connection, access to phone
- Note - virtual office hours have higher attendance than normal office hours, possibly use for normal courses as well. Note - probably need to require attendance though.
- Quiz/exam - use script with `at` to schedule when files are copied into class accounts and copied out.
- Exam review - campus/synchronous would be good.
- ISU video support - sometimes connection is good, sometimes not. Careful if planning anything that students need to be connected live for.
- Before term starts and at beginning - very clear that students need to have reasonable internet, access to a phone, etc.

Study Guide

- **Javascript in webpages**
  - **Quick start -**
    - use onclick attribute of html tag, put javascript code in the "" for the onclick, use id attribute on some other html tag, in the onclick code use document.getElementById("someID").innerHTML = "some string"
    - With document.getElementById("someID") can also set other attributes of the html element: src of img, css style (use camelCase for properties)
    - css properties commonly used: display (none/block)

        - Remember to use inspect element: looking at css, seeing which html elements have events attached, and looking for javascript errors in the console.
        - Put your js functions in <script></script> and then set onclick to be the name of that function with (). The script can go in the body after the html tag or in the head.
        - Debugging: have console open, run code, use console.log to trace if there is a syntax error.
    - **Where** - in script tag, inline to a tag, or in external js file (similarly to style that can be in any of three places)
    - **Functions/methods/attributes** - important ones - onclick, onsubmit, document.getElementById, *element*.innerHTML, *element*.style,
    - **Make sure** - have console window open when debugging page to make sure no js errors.
    - **Video** - javascript in webpages with demo and discussion
    - **Reading** - JS intro, events, examples, form validation, document object model, other common objects/methods, Steve Baker's CS 170, graphics
    - **Examples on CS server** - http://cs.indstate.edu/info/curiosities.html
- **Javascript versus C/C++**
    - **Why talking about** - CS 151 is javascript, CS 201, 202 are C/C++.
    - **Same** - much of the punctuation and keywords; basic logic of programming and debugging
    - **Different** - C is compiled (*compile* and then run, Javascript is *interpreted* [note - for C, any time you make a change you must save, then compile, then run]); C is *strongly typed* (must declare variable types, once declared type does not change); C has less built in (simple structures instead of objects, not as many builtin functions/methods); C file input/output is by default synchronous while javascripts is asynchronous (event-based, see example) - javascript in web-pages is *event-based*; printing is different (use printf for formatting output, C does *not* automatically convert anything for you)
        - Note: many programs in your computer were written in C/C++ (for example the operating system), compiled by the developers, and then packaged to be put on your computer to run.
    - **Strengths** - C is low-level and great when high efficiency or control are needed (OS, high-performance parts of games / networking, embedded systems, controlling the hardware); C forces you to understand how the CPU/computer really works (dealing with moving data around in memory, etc.); javascript is the language of the web (client-side programming in webpages, and can be used for server-side as well); javascript has more built-in and will have shorter programs (less typing)
    - **Video** - javascript versus C with demo and discussion
    - **Reading** - see http://cs.indstate.edu/info/getting-started.html#reference
- **General programming**

- ASCII code - the number that is actually stored for characters. a is 97, A is 65, Space is 32, 0 48.
- **Algorithm** - a set of instructions to solve a problem, a way to solve a problem. Note - problem should be extremely well specified, and so should the set of instructions (so computer or person can follow the instructions without having to make any decisions). **Note - computer science is really about programming and algorithms.**
- **Sorting** - given a list of numbers, the goal is to produce a sorted list. Normally we use “n” as the number of numbers.
  - *Selection sort* - scan through the list to find the smallest number (use a for loop) and put it at the beginning (swap with first item); then scan through the rest of the list to fine the second smallest and put it second; repeat.
    - What is the “running time”? How many basic steps, or how lines of code?
    - n steps to find smallest, n-1 steps to find second smallest, n-2, …, 1 step when we only one thing left unsorted.
    - Total: $n + (n-1) + (n-2) + \ldots + 3 + 2 + 1 = n (n+1) / 2 = n^2/2 + n/2$
    - We call that $O(n^2)$
  - Other $O(n^2)$ sorting algorithms: insertion sort, bubble sort, shell sort
  - **But, there are sorting algorithms (merge sort) that are $O(n * \log_2(n))$. For $n=10^9$, then that is 30 million times faster!!!**
- **Searching** - given a list of numbers and a number to look for, is that number in the list (or array)?
  - *Linear search* - look at each item in the array, return “yes” if item is the one we’re looking for. Return “no” if get through array without finding. Note (good) - doesn’t matter if the list is sorted or not, can always use linear search. Running time (bad) - roughly n steps, we call that O(n).
  - *Binary search* - Note (bad-ish) - array/list must already be in sorted order. Look at the middle location, test if number we’re looking for is larger, equal to, or smaller. Either we found it, or we can cut the list in half that we need to look at. Then repeat.
    - Running (good) - roughly $\log_2(n)$. **For $n=10^9$, this is about 30.** Why? Start with n, one step cuts the list (roughly) in half that we need to look at. So now we have roughly n/2 items to consider. One more step cuts the list in half again, so roughly n/4 items left. Eventually we only have 1 item left to look at. Number of steps, k, is such that $n * (½)^k <= 1$. $n <= 2^k$. k is roughly $\log_2(n)$.
    - Takeaway - if you’re going to to have lots of lookups, then sort your data. Note - there are other tricks as well; see data structures & algorithms (CS 201, 202, 303, 458).
- **Big O asymptotic running time** - we want a formula for how many basic steps an algorithm or a program takes depending on the “size of its input” (for sorting,

how many numbers). For sorting, given a list with n numbers, how many basic steps?

        - "Big O" basically takes the most important term (largest) in an expression for running time, and gets of rid constant that is multiplied.
        - Examples: $n^2/2 + n/2$ is $O(n^2)$, $100*n^3 + n^4/10 - n^2$ is $O(n^4)$
        - Note - for now you just have to memorize that selection is $O(n^2)$

- **Program design**
    - Split program up into pieces that are each easy enough to understand and test, and put them together so the program works. Each function should be at most one "screen" worth (so not more than, say, 20-30 lines of code).
    - If you are using the same hard-coded value multiple times in the program, define it as a variable or constant so you only have to change it once if you need to change it. (Reduces chances of forgetting and introducing errors later on)
- **Javascript hints**
    - Try out functions and such in the javascript console in your browser (right-click on page, click inspect element, then click console) or by running node in your terminal without a filename to run. When running node without a filename, type .exit to quit (or ctrl-d).
- **Javascript in the terminal**
    - We use node to run javascript programs in the terminal. Many of the basic functions are the same whether you're in the terminal using node or in a webpage. Some of the functions are different.
    - To run a javascript program, login to the terminal and run it like

      ```
      cd directory_with_javascript_file
      node javascript_filename.js
      ```
    - You can run the program with different arguments like this

      ```
      node javascript_filename.js argument1 argument2 argument3
      ```
    - In the javascript program, you use the arguments like this

      ```
      var nodeName = process.argv[0];  javascriptFilename = process.argv[1];
      arg1 = process.argv[2]; arg2 = process.argv[3]; arg3 = process.argv[4];
      var numArgs = process.argv.length; // would be 5 in this case
      ```
    - Note - use parseInt or parseFloat if needed to convert arguments.
- **Javascript methods, arrays, objects**
    - String methods: length, indexOf, slice, toUpperCase, toLowerCase, split, replace, charAt, charCodeAt
    - Numbers: scientific notation, hexadecimal 0x, numbers are 64 bit floating point, Infinity and -Infinity and NaN. Methods - toString, toExponential, toFixed, parseInt, parseFloat. Math methods - PI, round, pow, sqrt, abs, ceil, floor, sin, cos, min, max, random (returns between 0 and 1)
    - Arrays: create with [ ] in assignment, .length to get length, access with [0] and [k] for integer k, array methods - toString, push, pop, concat
    - Objects: create with {} in an assignment statement, name/value pairs for *properties* or functions, access properties or functions with . (can also access

properties with []), functions called “methods”. Note - there is more to using objects (object-oriented programming - creating our own new data types), but we don’t do that in this class.

- **Javascript basics**
    - Javascript is a general-purpose programming language! Html is NOT a programming language; it is a markup language.
    - Places for javascript: inline inside tag attributes, internal in script element, external in separate js file.
    - Good programming -
        - Write the smallest amount of code you can test, and then test it.
    - Reminder: to kill an infinite looping program in the terminal, ctrl-c
    - Basic syntax rules (common to most/many programming languages):
        - **Punctuation** that must be properly matched: (), [], “”, ‘’, {}
        - **Keywords** you are not allowed to use for anything else: for, while, if, function, do, break, switch, …
        - **Statements**: “does something”, end in ;
        - **Compound statements**: statements inside of { }, for example the statements of a function.
        - **Expression**: “has a value”, used in if, for, while, and other places
        - **Variable**: holds data, can save into (with =) or access in an expression, use var to declare local, without var is “global”
            - Warning: variables do not *have* to be declared. If you mistype a variable name in an assignment statement, javascript creates a new one (that is “global”). Note - if in strict mode, this doesn’t happen.
        - **Arithmetic operators**: + - * / % += -= *= /= %=
        - **Comparison operators**: < <= > >= != == === !==
            - Note: do NOT use = for checking equality
        - **Boolean operators**: (and, or, not) - && || !
            - Note: for A && B is true if both A and B are true. For A || B, is true if either A or B is true (or both). !A is true if A is false.
            - Note: in programming, 0 is false, 1 is true, and anything that is not 0 is often considered as true.
        - **Assignment and equality**: single = assigns to a variable, == or === tests for equality.
        - **Comments**: // to end of line or /* inside of this */
        - **Array**: use [] to create and to access, index starts counting at 0
        - **Parenthesis**: used for (a) grouping in math expressions, (b) definition of a function, with list of **parameters** in (), (c) calling a function, with **arguments** in ().
        - **Data types**: string (text, something in “” or ‘’, use + to **concatenate**), number, boolean (true or false, when converted to number true is 1 and false is 0), array, object. Automatic conversion between string, number,

boolean. String is text (not evaluated), access individual letters with slice using indexing like with arrays. Array is a list of items, in javascript they can be different types in the same array.

      - Integers - typically have a maximum and minimum possible value, depending on how much memory is allocated. In C or C++, typical integers go from -2^{31} to $2^{31}-1$ (roughly -2 billion to 2 billion).
      - Floating point - for numbers that have “decimal” part, or after the decimal point. Typically have a max/min and number of significant digits. In javascript all numbers are 64 bit floating point.
    - **A program**: a text file with proper instructions, that a computer can follow.
    - **Node**: is a program that can run javascript code. Javascript code also runs in the browser when part a webpage.
    - **Condition statement**: if (boolean condition) { statements; }
    - **For loop**: for(initializer; condition; update) { statements; /* loop body */}
    - **While loop**: while(condition) { statements; }
    - **Break loop**:
    - **Function**: a collection of statements with a name. Three steps are evaluating/passing **arguments**/**parameters** (argument value passed in, parameter is variable inside the function), running the statements in the function, passing back return value. Reason - can run same code multiple times without having to type it again, and also logically group together statements that do some specific task. Note: can have 0 parameters, can also not return anything.
      - Note on local versus global variables: variables declared inside of a function are “**local**” to the function, and are deleted when the function is done. Variables declared outside of functions are “**global**”.
    - **Escape characters**: To print a “ inside of a string, use \”. To print a \, use \\. Other escape characters - \n for newline, \t for tab.
    - **Best practices**: put “use strict” as top line in file.
    - **Syntax errors**: incorrect/unknown/misspelled/miscapitalized variable name or function, missing = in assignment or operator in expression, missing () in function call, missing “” or ‘’ for string
  -
- **Math**
  - Converting between bases: binary, decimal, hexadecimal, octal - see lab07_3, lab08_1, lab08_2, lab09_5, style2_bases
    - Algorithm from decimal to base X (X is 2, 8, 16, 10): remainder mod X, divide by X, repeat. Or, write number as multiples of powers of X.
    - Algorithm from base X to decimal: multiples of powers of X, and sum.
    - Algorithm to convert between binary and octal or hexadecimal: each 3 bits of binary is 1 octal digit, each 4 bits of binary is 1 hexadecimal digit.

        - Note - if you like binary the best, then you can convert always to binary. For example, to do base 10 -> base 16, you could do base 10 -> base 2 > base 16
    - Rules of exponents: see lab07_2, basic2_math
- **HTML style/CSS**
    - Methods: *inline* style attribute, *internal* style tag in head, *external* style sheet with link tag. Order of applying style: first external, second style in head, last style in tag.
    - Properties: background (-color), color, font (-family, -size), text-align, border, padding, margin, width, height (px, %), border-collapse and border-spacing (for table), list-style-type (disc/circle/square/none, for lists), text-decoration (none, for a)
    - Rules: tag, class (use . in css), id (use # in css), a (:link, :visited, :hover, :active)
    - Colors: rgb(255,255,255), color names, #ff00ff, what are the rgb and # codes for red, green, blue, white, black
    - span and div: often used around text just to give it style
    - Html box model: margin, border, padding, content. And background covers padding but not margin.
- **Using labCheck.html**
    - cp ~jkinne/public_html/cs151-su2017/LAB/labCheck.html ~/public_html
      chmod a+r ~/public_html/labCheck.html
      labcheck lab06
      Browse to http://cs.indstate.edu/~cs151xy/labCheck.html
- **Rules of html**
    - Make sure it validates as okay on the validator!
- **Using web browser**
    - Loading a local file (click File and then Open in most browsers)
    - Viewing page source (click View and then Page Source in most browsers)
    - Inspecting page (right-click and click “Inspect Element” in most browsers)
- **HTML**
    - Basic tags: html, head, title, body, h1, h2, h3, h4, h5, h6, p, a, img, br, hr, pre
    - Formatting: strong, b, em, i, small, del, ins, sup, sub, mark
    - Other: q, blockquote, cite, address, abbr
    - Comments: <!-- -->
    - Tables: table, tr, td, th, caption.   Attributes - colspan, rowspan.
    - Lists: ul, ol, li.   Attributes - type
    - Attributes: title, href, src, width, height, alt
    - Entities: &lt;, &gt;, &nbsp; &amp;, others
    - Layout and style: div, span
- **HTML pages on the CS server**
    - All files in your ~/public_html/ directory. See lab05 assignment.
- **How does a webpage get loaded**

  - If browsing to http://cs.indstate.edu/~cs15139/empty.html, then web browser connects to cs.indstate.edu and asks for the file /~cs15139/empty.html. Our server looks in ~cs15139/public_html/ for the empty.html file. The server sends the file back to the web browser, web browser displays it.
- Being able to connect to the CS server.
- HTML validation: https://validator.w3.org/
- **Terminology**
  - KB = kilobyte = 1,000 = 10^3 bytes, MB = megabyte = 1,000,000 = 10^6 bytes, GB = gigabyte = 10^9 bytes, TB = terabyte = 10^{12} bytes, PB = petabyte = 10^{15} bytes
- **Unix file permissions**
  - Each file has separate permissions for the user that owns the file, users in the group that the file belongs to, and everyone else. For example, if user cs15139 does

    ```
    cd ~
    ls -l hello.txt
    -rw-r-----+ 1 cs15139 class     6 May 24 09:28 hello.txt
    ```

    This means user cs15139 can read and write the file, every user in the class group can read the file, and all other users do not have any access.
  - For directory permissions, r is for listing the contents, w is for creating a new file (or removing a file), x is for "going into" the directory.
- **To make your life easier - more shell stuff, file management, local text editors**
  - rc files - are loaded when you login, either .cshrc, .bashrc, .tcshrc
    - In .cshrc add lines
      alias rm 'rm -i'
      And also for cp, mv
  - shell script - put in shell commands, make executable, first line #!/bin/tcsh, then run by being in the directory and typing ./shellScriptFilename
  - rsync - useful for backup/copying files
    - rsync -av SOME_DIRECTORY BACKUP
  - prompt before overwrite - use -i with cp, rm, mv, can set as default using alias in rc file
  - Files - use sftp program to download files to your local computer - FileZilla, WinSCP, sftp
  - Local editor - text editors on your local computer - you can download emacs or vim, or use atom or Eclipse (cross-platform GUI editors).
- **Shell/terminal commands**
  - man - get help on a command (man stands for manual)
  - cd - change directory
  - ls - list directory contents. ls -ltrh to list details, sorted by time in reverse order (most recent last), with human-readable size (in KB, MB, etc.)
  - cp - copy file;   mv - move file
  - rm - remove a file
  - mkdir - make directory;   rmdir - remove directory

  - pwd - print working/current directory
  - clear - clear screen
  - chmod - change file/directory permissions. chmod go-rwx hello.txt   takes away permissions from everyone except the user that owns the file.   chmod a+rx * gives read and execute permissions to everyone to every file in the current directory.
  - more - list file contents, use q to quit
  - less - list file contents, use q to quit
  - See “Using Linux” section at http://cs.indstate.edu/info/getting-started.html
- **Shell/terminal special symbols and control**
  - Up arrow goes to last run command. Tab tries to auto-complete.
  - >
  - <
  - &
  - | pipe symbol, sends output of command to the input of the next
  - ~ is a shortcut for your home directory. ~jkinne is a shortcut for jkinne’s home directory
  - .. is a shortcut for going up one level in the directories.
  - . is a shortcut for the current directory.
  - * when there should a filename means “everything”. Can also use * as part of a filename that means “anything” for that part. Example: ls -l *.txt
  - ctrl-c (holding control key down, and pressing c) - kills the current process
- **More useful shell commands**
  - head, tail - display the first few or last few lines of a text file
  - df - display disk free information (df -h to display in nicer format)
  - du - display disk usage information for a directory (du -h -d 1)
  - free - display memory usage information (free -h)
  - top - display information about running processes (like task manager or activity monitor)
  - grep - search a text file for a string (grep “something” file.txt)
  - wc - word count, gives # lines, words, and characters (wc file.txt)
  - finger - see who is logged on to the system
  - passwd - change your password
  - date - see the current date and time
  - cal - calendar
  - bc - binary calculator
  - locate - find files that have been used in the system. Sometimes doesn’t find a file. Example: locate -r /filename.txt$
  - find - search directories for files. Does actually walk the directories, so finds files locate misses. But is slower. Example: find ~ -name README.txt
    Example: find ~ -mtime -2
    More help on find.
  - stat - information about files/directories - permissions, access dates, etc.

    - cat - concatenate, but can use to print a file, like cat hello2.txt
    - More commands - are in /bin, /usr/bin, /usr/local/bin, /usr/games
- **Text editors** - for each you can open a file by typing the command and then a text file name (e.g., type pico hello.txt).
    - **pico** - use the key-codes listed on the bottom to get around, ^ on the bottom means hold down the control key
    - **vim** - use :w to save/write, :q to quit, :q! to quit without saving, i to enter "insert mode", ESC to exit "insert mode". vim website (can be downloaded for windows, normally builtin on mac, linux)
    - **emacs** - use ctrl-x ctrl-c (hold down control while pressing x, then hold down control while pressing c) to exit, ctrl-x ctrl-s to save, sometimes a message appears on the bottom that you have to answer. emacs website (can download for windows, mac, linux to edit files locally - often builtin on linux and mac)
    - **Other GUI editors** - atom or eclipse (note, on mac os x need to install JDK 8 first) are cross-platform and open-source. atom is more geared towards web-development, while eclipse is often used for java.
- **Other programs**
    - File transfer: FileZilla, WinSCP

Notes (most recent first)

- Plan for the semester (based on Steve Baker's normal CS 151 outline and a review of ACM standards mapping to ISU courses)
    - See outline in syllabus …
- Sometime soonish
    - Jeff needs to do:
        - check with Manufacturing Technology and other majors that require CS 151, are there any slight changes that would benefit them
    - Video to show people's examples from hw4 and answer questions.
    - Lecture - programs to memorize.
- 8/11 Fri
    - Lecture - javascript in webpages. Not on the final exam, but you want to know this stuff, and perhaps a preview of CS 170. If you want to work on this, then go through the W3Schools tutorials and copy/paste the code into your account on the CS server, making sure it works. As you get further in doing more examples come up with something you'd like to do that is very similar to some of the examples.
    - Lecture - differences between javascript and C - head's up going into CS 201. If you want to work on this, then start reading through a C tutorial for beginners, copy/paste some of the examples on the CS server with your CS account, make sure they run/compile, and just work through them.
- 8/8 Tue
    - Everything is graded and final grades are in. I will put a few videos mentioned above up this week sometime. I'll write to the class when that happens.

  - Your cs151xy accounts will be deleted sometime next week. To save your files to your own computer, use one of the file transfer programs mentioned above.
  - The files in my account and the google doc will stay online "forever".
- 8/7 Mon
  - Final exam phone calls.
- 8/6 Sun
  - Final exam taken between 8/4 and 8/6
  -
  - Jeff traveling, limited or no email contact.
- 8/5 Sat
  - Jeff traveling, limited or no email contact.
- 8/4 Fri
  - Jeff traveling, limited or no email contact.
- 8/3 Thu
  - Quiz6 due by 11:59pm. Covers new topics on javascript from 7/31.
  -
  - Jeff traveling, limited or no email contact.
  -
  - Exam3 has been set to transfer into your directories at your start times. Good luck!
- 8/2 Wednesday
  - Lab15 and short-quizzes on lectures due.
  -
  - Note - quiz6 will not be graded before you take the exam. But the answers to the questions will be accessible to you after 2am Friday at http://cs.indstate.edu/~jkinne/cs151-su2017/QUIZ/
    The files have read permissions off now until 2am Friday. One of the files has the questions, and the other has the answers. They will be in a different order, and sometimes with different numbers than the questions that each of you had.
  -
  - Note - additional videos I'll make - listed in the "Sometime soonish" bullet point above. Possibly on 8/7.
  -
  - Lecture - quiz5 and exam2 answers part 1 and part 2.
  -
  - Lecture - quiz questions trace and algorithm. This is a video showing a new type of quiz/exam question - "trace the code" and "run the algorithm", and what I expect for the answers. For quiz6, you will have a "run the algorithm" for selection sort, and will have a "trace the code" for code with a for loop. For exam3, you will have a "run the algorithm" for binary search and will have a "trace the code" for code with loops, ifs, and functions.
- 8/1
  - Lecture - algorithm, big O, linear search binary search.

        - Reading assignment: binary search at tutorialspoint. See also - wikipedia or *Introduction to Algorithms*.
        - See also linear search
        - You should be able to: given a list of numbers, show what happens with either binary search or linear search. Also, say what the number of steps would be for each. For binary search, be able to say what low, mid, high are as binary search runs.
        - Lecture short-quiz - there isn't one for this lecture. There will be a question on the final exam.
    - Lecture - algorithm, big O, selection sort.
        - Reading assignment: selection sort at tutorials point. Read more - selection sort at wikipedia. Or even better - selection sort in the book Introduction to Algorithms (any edition)
        - Good quiz/exam questions: given a list of numbers, what does the list look like for each iteration of selection sort. For a given size of a list of number (e.g., for n=10^6 numbers), roughly how many basic steps does it take to run selection sort?
        - Lecture short-quiz - there isn't one for this lecture. But there will be a question on the final exam.
- 7/31
    - Lecture - lab15. The last lab assignment. Due 8/2. Watch the other three videos (functions review, then arrays, then objects) first.
    - Lecture - javascript objects. Examples - person data, point. Short quiz due 8/2. Lab15 due 8/2.
        - Reading assignment - see links in objects section up above.
    - Lecture - javascript arrays. The most basic data structure in most programming languages. Examples - random numbers, findMaxMinAvg. (Another example we could have done - chat2). Short quiz due 8/2. Lab15 due 8/2.
        - Reading assignment - see links in array section up above.
    - Lecture - functions review. Examples - findMaxMinAvg, area. Other examples we could have done - (makeLine with multiple parameters, flipCoinsUntil). Short-quiz due 8/2 (incorrectly says 8/1 in blackboard). Lab15 will be due 8/2.
        - Steps: (1) arguments to function evaluated, (2) arguments copied in to parameter variables, (3) function code is run until the end of the function or the first return statement, (4) value from return statement is copied back to where function was called (and saved into variable if function was part of assignment statement).
        - Remember: evaluate arguments, copy into parameter variables, run function, copy return value back out.
        - Note - function can have 0, 1, 2, etc. arguments/parameters, but the number of arguments when calling the function should match the number of parameter variables on function definition.

        - Note: variables declared inside of a function are “local” - only can use inside of the function.
- 7/30
    - Grading notes…
        - For programs, if they don’t run they are worth 0 points. Make sure they at least run.
        - Something like 2.3e4 is scientific notation. That was covered in one of the math videos.
        - For both quiz5 and exam2, your answers file with some notes on what was marked wrong (X for wrong) are in your ~/QUIZ/ and ~/EXAM/ directory.
    - After reviewing what we’re supposed to get done, there will be a few new topics covered this week and on the exam. I’ll make up the schedule later tonight. The new topics -
        - Javascript topics still to add: arrays, object with properties/values, special values (None, NaN, Infinity), selection sort, big O running time (linear versus quadratic)
        - Vocabulary: global variable, local variable
- 7/29
    - I'll grade everything (including exam2) tonight or tomorrow and be in touch with what I think people need to work on. I'll check what we're supposed to finish to see if there is anything we missed that we should fit in this week. If people want me to look back at any labs or HW's let me know.
- 7/27
    - Exam2 review - will be similar to quiz4, quiz5, but longer. Focus is on javascript, but of course you cannot forget the things about unix, terminal, etc.
        - For review live session, ask questions on youtube live, or send email. Ask for more examples on if, for, while, numbers, strings, syntax errors, functions. Or ask to go over anything.
        - Programs to have **memorized by heart** - study/practice until you can type them in without consulting anything else.
            - Check that there are enough command-line arguments, and convert to float or int if needed.
            - Do something a certain number of times (specified with command-line argument, or hard-coded into the program).
            - Split a string at a character that is searched for.
            - Putting together a string from multiple parts - use +, might have to some other thing.
            - if/else if/else if/etc./else to do something different depending on the value of a string or number.
            - That is how far we got!
            -
            - Next would be -

- Two nested loops - addition table, printing a rectangle.
- Properly use all of the arithmetic operations and Math.pow, Math.sqrt, to evaluate math expressions (e.g., in the javascript console).
- New (for final exam) - For loop to add up an array of numbers. Note - use .length
- New (for final exam) - For loop to find the maximum or minimum of an array of numbers. Note - use .length

- 7/25
  - Quiz4 grading. The # points listed in the quiz was wrong. There were 5 ⅔ points from fill-in-the-blank, and I am setting the programs to 2.5 points each for 5 points total for the programs. I'll make the total for quiz4 as 10 points, and won't take off for the first fill-in-the-blank that is wrong.
    - Punctuation for order of operations: answer was ( )
    - Punctuation for string constant: " " or ' '
    - For those who took the quiz, a copy of your answers is in your ~/QUIZ/quiz4-graded.txt. Fill in the blank that were wrong are marked with X, and #17 and #18 have a brief reason why points were taken off, if any were taken off.
    - Lecture - answers for quiz4. Answers to fill-in-the-blank only. Note - no lecture short-quiz or lab assignment.
- 7/24
  - Lecture - javascript hints - the console. Quick video suggesting you try out the javascript console to test your knowledge and help study. No lab assignment or short quiz. *Uploading.*
  - Graded - lab12, lab13, hw4.
    - On lab assignments, if it looked like you started a problem and didn't have it, I left a comment in your code somewhere of the next thing you need to do. Check the comments in your code.
    - On all lab assignments, note that you need to have the usage statement correct. The first test I always run is running your program without any arguments to check the usage statement.
    - HW4 - questions for me to answer from your hw4.txt
      - cs15101 - setTimeOut and show multiple time zones or convert between time zones
      - cs15105 - try/catch
      - cs15104 - use multiple colors in gradient
      - cs15107 - error with errors.join in example file
      - Note - javascript is NOT java.
      - Make a link to Jeff's directory.
      - Links/resources for more in depth with html/css/javascript
      - Lab13 last 3 parts.

        - HW4 - examples for me to explain from your hw4_example_cs151xy.js files
            - cs15107 - password validation
            - cs15111 - information about browser
            - cs15114 - collecting links of images in the page and displaying them
            - cs15120 - using Date to display date/time
    - Graded - discussion / yuja chat / lecture short-quiz for past two weeks. Remember that you are supposed to login and say something in the yuja chat at least once per week, and post something (or reply) in the blackboard discussion at least 3 times per week, and watch the lecture videos and complete the short-quiz for each by the stated due dates (generally the Wednesday or Saturday following when it is released).
    - The schedule has us doing the second exam the end of this week, and a weekly quiz due Wednesday. We will stick with that schedule. For the weekly quiz you can pick any time tomorrow or Wednesday to take it. For the exam you can pick any time Wednesday, Thursday, Friday, or Saturday. You will get 45 minutes for the quiz and 90 minutes for the exam.
        - If you do not ask for a quiz time before tomorrow morning, I will set them by default to Wednesday at 6pm.
        - For the exam, if you don't pick a time before tomorrow at 8:30pm I will set them by default to Saturday at noon.
    - lab14 has been released, due Wednesday. The lecture short-quiz for the last two lectures is also due Wednesday.
        - Lecture - lab14
    - On yuja chat this week - all 9am/8pm through Saturday, except - Tuesday and Thursday morning and Friday evening.
- 7/23
    - FYI, my internet isn't working well tonight. I won't be on chat. You all can chat if you have questions. There will be a lab14 posted tonight or tomorrow that will be due Wednesday. The second exam will be everything up until lab14, and it will be the end of this week. I'll send more details tomorrow.
    - Lecture - math methods. Ways to specify numbers (scientific notation, regular, hex), limits of javascript numbers (precision, size), converting between int/float and string, builtin in math methods (e.g., sqrt, pow, floor). Examples - flipping coins, pythagorean theorem.
        - Reading assignment - see links in study guide above.
        - Lab assignment - lab14.
- 7/21
    - Lecture - string methods. Builtin string methods. String methods are often used for parsing strings. Examples - parse phone # for area code, split name into first and last, simple chat. *Uploading...*
        - Reading assignment - see links in study guide above.

        - Lab assignment - lab14.
- 7/19
    - HW4 has been released, due Saturday. Part of the assignment is to talk to Jeff sometime this week, so write to me to get a time. Another part is to pick a "buddy" to work with on the assignment.
    - Weekly quiz - will be taken between Thursday and Saturday. Write to me to say what time you want to start the quiz. For those I don't hear from I'll set the start time to Saturday at noon.
- 7/16
    - Yuja chat - I'll be on this week each night 8pm (sorry, no mornings this week). Remember you're supposed to check in there at least once per week.
    - Lecture - lab13. Lab13 released, due 7/19. Lecture video describing it will be uploaded overnight.
- 7/14
    - Lecture - loop, function examples. Some more examples using what we know - ASCII art, addition table. Note - program design - make small parts that are easy to understand and test, and put them together.
        - Reading assignment - …
        - Lab assignment - lab13.
- 7/13
    - Lecture - condition, function examples. Some more examples using what we know - unit conversion checker, evaluating binary operators. Use strict, escape characters. *In progress, will be uploaded later today/tonight.*
        - Reading assignment - strict mode, escape characters (in part on strings).
        - Lab assignment - lab13, will be due 7/19. *In progress, will be uploaded later today/tonight.*
- 7/12
    - Next set of lectures - javascript in webpages, next week.
    - Lecture - javascript with node and arguments. Running javascript programs in the terminal with node, and passing arguments to javascript programs in the terminal. Take previous in-class examples and make them use command-line arguments.
        - Reading assignment - notes on javascript in our study guide.
        - Lab assignment - lab12 - has been released, described at the end of the video, due 7/15 by 11:59pm.
- 7/10
    - I'll be on chat this week Monday 8pm, Tuesday 9am, Thursday 9am and 8pm, Friday 9am and 8pm, Sunday 8pm. Note that you're still supposed to post at least 3 times in the discussion, and participate at least once in the chat.
    - Note that right now we're first doing basic programming concepts before we do "fun" stuff with it in web pages. There is no way to do the fun stuff if we don't have the logic and debugging skills we'll need. Now is the time to be extra diligent, patient, and positive-thinking.

    - Lab11 graded and grades for week 5-6 discussion and chat were updated. Week 7 discussion and chat threads have been created. For lab11, if you didn't get all the points I left a comment in whichever files were wrong with a suggstion.
- 7/6
    - Note - youtube playlist has been flipped, with oldest first in the playlist so that multiple videos can more easily be seen one after another.
    - Look at your "total grade" and each of the grade categories in blackboard. There is also a "Letter grade estimate" now. Remember that you need a C in the class to be able to take CS 170, CS 260, or CS 201. Those who didn't take the first exam have an F right now. Everyone has something they could do better. Most of the people below B have an entire part of the course they haven't done right (e.g., no attendance, or poor on HW's, quizzes, or exam). Check the categories to see which you need to work on.
    - Exam1 is graded, grades in blackboard. For those who took the exam, your answers file was copied into your account at ~/EXAM/ with marks in the file on what you got wrong. There is an X before each question you missed, and XX's on the first 3 (the html files) with brief explanation of what you didn't do right. There were a total of 24 points, and each X was -⅓. You can check that I added up all the X's correctly and got it into BB correctly. If you have a question about one of your answers, spend at least a few minutes thinking about it before asking.
        - Note that if you better on the second and third exams than the first one, I'll drop the first exam grade. BUT, that only ever happens for people that do something different in terms of their work habits. Note that the material for the second half of the course is slightly more abstract/difficult, so definitely don't take a break from the class right now!
        - Lecture - exam1 answers. Watch this video to see what the answers were for the fill-in-the-blank questions. Check the answers in the video against your answers to see whether the answers in the video are better than what you said. Note - no lecture short-quiz for this video. *Uploading.*
    - Reminder - post into blackboard discussion at least 3 times per week, say something on the chat at least once per week. We'll keep counting those as part of attendance/participation, together with the lecture short quizzes.
    - Graded lab10. Note that if I cannot browser to your pages, you need to fix the permissions for your directories and files. If you didn't receive full credit then at least one of your parts wasn't correct. If you don't know why, ask in the chat or discussion.
- 7/5
    - Lectures on programming concepts - reading assignment is w3schools javascript tutorial from section Syntax to section Functions.

        - Lab11 has been posted, due Saturday by 11:59pm. Video will be posted in the morning showing how to get started and how the labcheck will work for this one.
        - Lecture - using labcheck for lab11 javascript. No lecture short quiz. Watch this video before you get started on lab11
        - Lecture short quiz has been posted for these 4 videos. Note that it is a single quiz that can only be taken once. So type and record the quiz question answers as you watch, and then take the quiz after you're done with all four. Due Saturday be 11:59pm.
        - Lecture - programming concepts part 4 - if, for, while, boolean expressions
        - Lecture - programming concepts part 3 - functions - defining and calling.
        - Lecture - programming concepts part 2 - data types in javascript - number, string, boolean, array, object.
        - Lecture - programming concepts part 1 - statements, expressions, variables, program (a text file with instructions), sytnax errors, console.log, using node to run javascript programs.
- 7/5
    - Catching up on grading, will send a message when it's all done.
-
    - Note on chat and discussion - have just extended last week's threads to go through this week as well.
    - Yuja discussion feed - I'll be logged on this week today 8pm, Wednesday 9am and 8pm, Thursday 8pm, Friday 9am, probably Saturday and Sunday 8pm (will depend on internet connection where I am staying those days).
- 7/1
    - I'm back from being away. Will grade the exam tomorrow, put in an estimated total grade for everyone, and say something about that.
- 6/28
    - Lab10 is up now and is due Saturday by midnight. I'll check in on it when I get back on Saturday. Note that there is no labcheck this time, you just need to make sure your files do what is asked in lab10.txt.
- 6/27
    - **Scheduling note - I will be out of email contact (and no new content/assignments) from 6/28 10am through 7/1 noon.**
    - Reminder - labs and hw's you can fix for late credit. Just let me know. (I still have to check a few people's who asked me and let them know.)
    - Lab09 graded with grades in blackboard.
    - Hw3 graded with grades in blackboard. I left a note in your ~/hw3.txt file in your account about where your grade came from.
    - Lecture - hw3 improvement for someone's game. Using style to make different parts of a table look different ways. This is not required, it's just for fun.
    - Lecture short-quiz for the introducing javascript videos is due Saturday.

    - Lecture - introducing javascript part 2. W3schools javascript tutorial - Where to, Output.
        - The goal right now is to get enough of the rules in our mind to be able to use javascript. There are a lot of rules, they must all be followed!
    - Lecture - introducing javascript part 1. Reading material to accompany - w3schools javascript tutorial - tutorial home and introduction. *Will upload over night.*
        - Note: our goal in studying javascript is to (a) study programming, (b) do fun/visual/useful stuff.
- 6/26
    - Note - there is a new lecture short-quiz for these videos, that is due 7/1. Remember - check the quiz questions before watching the videos, then it should be easy to watch the videos and pay attention to what the questions are asking for.
    - Lecture - things to make your life easier - other GUI editors. This gives a look at two text editors that are GUI editors that you can use on your local computer - atom and eclipse. You would download files from the server using an sftp program (like FileZilla), work on the files with your GUI text editor, and then upload to the server when you're done.
    - Lecture - things to make your life easier - backup files. Using the rsync command to copy directory contents into a backup folder. How to create a shell script file that can contain multiple rsync commands to backup multiple files/directories. And a reminder that you can use an sftp program (like FileZilla) to download files to your local computer (and send them back to the CS server).
    - Lecture - things to make your life easier - confirm before delete. How to set things up so that you will be asked before deleting files (as a result of rm, cp, mv, rmdir). This includes where some configuration files are kept for you.
- 6/25
    - For the yuja chat this week, I plan to be on Monday 9am and 8pm, Tuesday 8pm, Wednesday 9am, Saturday 8pm, Sunday 8pm. This week we begin javascript.
    - For those who don't pick a day/time to start the exam I'll have it start Wednesday at 6pm for you.
- 6/24
    - Reminder - you need to write to me to ask for a time to start the exam - any time from 12:01am Sunday through 11:59pm Wednesday is fine.
    - Hw3 due date changed to 6/25 by 11:59pm.
- 6/23
    - Quiz3 grading …
        - If you didn't take the quiz 0. If you didn't put the answers in the file it asked for, -1. If either of your html files did not validate as correct on the html validator, -1 each. -.5 for each other mistake in the html files. -.5 per wrong other answer (#s 1 - 18).

        - I copied your answers file back into your QUIZ/quiz3/ directory with marks on it for what was taken off for.
- 6/22
    - Write to me to ask for a time to start the exam - sometime Sunday to Tuesday.
- 6/20
    - Lecture - converting between bases part 2 - more, including convering from decimal to base X (for X = 2, 8, 16). *Uploading.*
    - Lecture - converting between bases part 1 - basic theory of different bases for numbers (binary, etc.) and converting from base X to decimal. *Uploading.*
    - *Note that the weekly quiz this week will have some questions on converting between bases.*
    - Course admininstration videos - reminding you what you need to do. These are not required (no lecture short quiz), but if you have missed points on anything you should watch them. They are short.
        - Video - administration of course - this week's quiz, and first exam - short video of just about 6 minutes reminding you of things you need to do, and talking about this week's quiz. Note that you MUST email me when you want to take the quiz, or you will get a 0 for it.
        - Video - administration of course - reminder about labcheck, labCheck.html
    -
    - Currently for you to do: old labs that weren't correct, lab09, hw3, **email Jeff when you want to take the weekly quiz, email Jeff when you want to take the first exam**.
- 6/19
    - **First exam**
        - Content - everything we have done so far. See the study guide above. Roughly, we have learned how to use unix, deal with text files, and the basics of html and css. (In the next part of the course we'll get into programming with javascript.)
        - Types of questions - everything that has been on the quiz, and selected parts of the labs.
        - You can take the exam between Sunday and Tuesday (from 12:01am June 25 to 11:59 June 27). You will have 90 minutes to finish the exam. You will write to me to ask for a time to start the exam. This will be handled just like the quiz - see the next set of bullet points about the weekly quiz.
        - Sample exam - I may release a sample exam sometime this week.
        - Videos - I will make a video about converting between bases, and I will make a video explaining how we will do the quiz and exam this time. If you think you'll need any other videos to help with, or there is something you are interested in, let me know. For this week, I will only make videos of things you're interested in.

        - Remember that exams collectively count for 45% of the points in this class.
        - According to the policies for CS courses, there will be no makeup exams. If you do not take the exam, you will have a 0 for it.
    - **Weekly quiz from style/css - we'll do this quiz slightly differently.**
        - You decide when you want to take the quiz on Wednesday or Thursday (from 12:01am Wednesday to Thursday 11:59pm).
        - Write to me sometime tomorrow (Tuesday) to tell me when you plan to start the quiz.
        - I will make sure the quiz is in your account by the time you have requested.
        - You must have your answers saved in your cs151 account within 45 minutes of your start time (I'll check the last-modified time).
        - By turning in the quiz, you agree that you have not communicated with anyone else about the quiz. You can feel free to consult your notes and use the internet for reference material. But you must not communicate with anyone about the quiz until after Thursday.
- 6/18
    - Yuja-chat/discussion-feed - this week I'll definitely be on Monday 9am, Tuesday 9am and 8pm, Friday 9am and 8pm. I'll possibly be on Wednesday 8pm, Thursday 8pm, Saturday 9am; I make no promises on those, I'll check it if I can.
- 6/17
    - **Hw3** has been released and is due in one week (Saturday June 24). You will make a website that looks something like a game. You must get approval from me first, so get to work at least a few days ahead of time in picking your game.
        - Games already taken: bingo, Zingo, Monopoly, Snooker, Snakes & Ladders,
        - Game NOT allowed: chess, checkers, tic-tac-toe, hangman
    - **Lab09** has been released, is due Wednesday. More practice with tables and style. 5 parts, 11 points total. Get started early.
- 6/16/2017
    - Reminder - before you watch a video lecture, check the short-quiz for it for what the question will be. You can keep that in mind as you watch to make sure you get the question right.
    - Hw2 graded. Remember that if your index.html does not pass the html validator at https://validator.w3.org/ then you get 0 credit. I will require that as well on future lab and hw assignments. I will look at the hw2's again on Sunday afternoon to see if any have been fixed.
    - Lab07 graded. Note that your html file must pass the labcheck, or it counts for 0 credit. This is to drive home the point that you need to meet a specification. The ability to know when your solution is wrong, and troubleshoot fixing it, is a very important CS skill!

        - For those with incorrect solutions, I left notes in a file lab07.txt in your home directory with some things I noticed were wrong.
        - Common problems: not having spacing correct (should have a space after commas and periods but not before them), typo in one of the image links (there was an l that people thought was a 1), only specify width for the images, don't give style/css at all (in particular don't try to make the page look like it is with a black background) for lab07, typos of words.
        - I will take a look again at which ones are correct on Sunday around noon. If you are correct by then, I'll count it as correct on time.
- 6/15/2017
    - Scheduling note - new lectures and lab08 due June 17 by 11:59pm.
    - Reminder - your work flow - (1) watch my videos, (2) read w3schools (or other reading if I suggest it) and do some of the "Try It Yourself"s, (3) do the lab exercise, (4) do the hw assignment. And repeat. For the lab and hw, google search for particular things as needed (e.g., it's okay to google something like - html css text color).
    - Hw3, next weekly quiz - will be due Wednesday June 21
    - **Lab08** - based off of style and css videos. Released and due June 17.
    - Lecture - style and css in html part 2 - HTML sections on Styles, Colors, CSS, styles in previous HTML sections we looked at. Looking at more properties that are commonly used.
    - Lecture - style and css in html part 1 - HTML sections on Styles, Colors, CSS, styles in previous HTML sections we looked at. Focus on showing a few style properties, different ways to give style (attribute inline, style tag internal to page, style file external to page), and how to play around with different styles in the browser (inspect element).
- 6/12/2017
    - Note - first exam will be delayed by 1 week from original schedule so we can include css/style, and so you get extra hw and quiz practice.
    - Yuja - other people at ISU have said the video feature in yuja does not work well with multiple people broadcasting. We will try again at some point with one person at a time broadcasting.
    - On due dates - will modify the schedule slightly so you have fewer due dates to remember. I will make everything due either Wednesday by 11:59pm EST or Saturday by 11:59pm EST. This will mimic a typical tuesday/thursday class where you'd have two days per week to pay attention to. This also means you likely will be most interested in live-chat/discussion on Tuesdays and Fridays (because there is no way you would wait until the day something is due to start it, right?).
    - Hw2 - gave it a quick look, and nobody had it right. Note that 0 credit is given if doesn't load in a web browser (because not present or incorrect permissions), or if doesn't pass w3c validator. See grading standards in the hw2.txt file for how

the index.html is graded. A reasonable example for you to look at - http://cs.indstate.edu/~cs15103/ but note that it doesn't pass the validator

        - On time due date extended to Wednesday 11:59pm, you may want to ask in live-chat on Tuesday about your assignment if you aren't sure what you're doing wrong.

    - If you haven't talked to me yet for hw2, I will give you those 2 points if you talk to me this week.
    - Lab07 - based on the results from lab06, I am extending the due date to Wednesday by 11:59pm.
    - Lab06 - graded and now open up for fixing mistakes. Make sure to watch the video about labCheck.html and use it to fix your lab06. Consider this as a requirement to pass the course. Note - do NOT have any style or div's in lab06 or lab07. We haven't covered those yet, and they will only make your assignment incorrect.
    - Week 2 weekly quiz - a few missed taking it. I have enabled it again, those who take it late will have it count as 50% (so a perfect score on the test will count as 50%). I'll close it again by Wednesday 11:59pm.
    - There is an issue with the permissions in your accounts. They are automatically getting reset so the ls -l isn't displaying correct information. For now, I changed lab07 so it lets the check continue even if it thinks there is a problem. And I'll not count the points for that part as I grade them. As long as you can browse the pages in your web browser that's fine.
    - For this, week I'll have live-chats: Monday 9am, Tuesday 9am and 8pm, Wednesday 9am, Friday 9am and 8pm, Saturday 9am and 8pm, Sunday 8pm.
    - Change in plans for live-chat - I will hold the live chat every day at 9am and 8pm, as long as that time works for me. I'll put a list on Sunday of the days we'll do it each week. You'll still only need to participate once. I will do this week like this and see which days end up getting the most use.
    - I am back in touch and getting caught up. Will first grade everything that is past due. New videos to be uploaded late tonight.
- 6/8/2017
    - Scheduling note - I will be out of action from 6/8 - 6/11, and also from 6/28 - 7/1. For both date ranges, we won't have anything due on those dates, and you can expect to probably not hear from me.
    - Note: for the discussion, you should be participating at least 3 times per week. That means you comment or ask a question at least 3 times. The reason is so that you will try to check it multiple times per week (to think about things yourself, and also check on others). For the chat, you just have to show up to one of those per week.
- 6/7/2017
    - Scheduling note - exam scheduled for 6/18-6/19. We can discuss on live-chat at 8pm, those who take part then.

    - Lecture - using the terminal/shell in unix part 3 - more commands and what they do (the list “More useful Shell Commands” above), using FileZilla or some other ftp program to transfer files, more about file permissions.
    - Lecture - lab07 assignment. The video just goes over the expectations for the lab07 assignment.
    - lab07 has been released and is due Sunday June 11 by 11:59pm.
    - **Note that anything in your public_html can be viewed by other students. You should NOT just copy someone else’s html. That is cheating. If I can tell you have done this, it is a 0 for the assignment at a minimum; it is within my rights to give an F for the course for doing this. DO NOT COPY other people’s code. The rules are - “no electronic copying of any kind”. It is okay to copy any of the files I have given you, but that is. See http://cs.indstate.edu/info/policies.html#cheating**
- 6/4//2017
    - Hw2 released - see HW folder. Due June 8 by 11:59pm.
    - Week 2 quiz released. Due June 8 by 11:59pm. **Note - once you start you need to finish within 45 minutes, and you can only take it once. And you cannot close the quiz once you start it.**
    - Survey released, please fill it out, to indicate times you are available to talk one-on-one. There is a link for Surveys now on the left-hand side in blackboard.
    - Note: week 1 quiz regraded, fixed some mistakes in the auto-grading. For quizzes, try to answer what you think I’ll type in for the auto-grader to check. For example, if the question is “What is the shell command for copying a file?”, then cd is a better answer than “cd is the command”.
- 6/3/2017
    - Grading notes … Everything that was due is graded. For lab05, if the html pages were not viewable (if you forgot to do chmod) you got 0 credit. I will check them again next week.
    - labcheck for lab06 - it now will check that your html files are correct. It isn’t foolproof, but normally will be correct. To get the benefit of this, see http://cs.indstate.edu/~jkinne/cs151-su2017/LAB/labCheck.html first. That tells you to copy that html file into your public_html directory. Once you have that, run the labcheck for lab06 (assuming you’ve started on it). If it finds any parts that are wrong, it tells you to browse to the labCheck.html page to see your page compared against the correct one.
    - Scheduling notes. hw2 and the second weekly quiz will probably be released by 11:59pm June 4, and will have due dates pushed back as well.
    - Answering questions and things that were asked in hw1 -
        - How many bytes in a PB. It’s 10^{15}
        - Note that the # of processors listed in /proc/cpuinfo is the # of hyperthreads. That is twice the number of cores. There are 2 CPU’s in the server, so half the total number of cores is the number in one of the CPU’s.

        - It seems people would like a video explaining the answers for hw1 and the first weekly quiz. Okay.
        - To have quicker access to my directory, you can do this

          ```
           cd ~
           ln -s ~jkinne/public_html/cs151-su2017 jkinne-cs151
          ```

          Now if you will see a link jkinne-cs151 in your directory that you can use to get into my directory. You could name this whatever you want (e.g., just j instead of jkinne-cs151). Note that if you cd into the directory, cd .. won't get you back to your directory; you'd have to cd ~ to get back yours).
        - To get the total HDD space, you just have to add up all the lines in the df output. It is possible to automate that using other commands.
        - For the #2 questions you had to say what your answer was, not just give the result of running the command.
        - /etc/passwd is a text file that contains one line for each account on the system. wc gives the total # of lines of a file as one of the numbers it outputs.
        - If you skipped #3 I took off 2 points.
- 6/2/2017
    - Lecture - HTML basics part 2. More tags, attributes etc.
        - Extra resource: HTML at w3schools, the following chapters - Formatting, Quotations, Comments, Tables, Lists. And over skipping a bunch of sections, we'll look at - ComputerCode, Entities, Symbols,
          Note: skipping and coming back to slightly later - Styles, CSS, Links, Images, and a few others.
- 6/1/2017
    - Lecture - HTML basics part 1. Basic HTML rules, basic set of html tags, html validator, common mistakes and how they show up in the validator and browser.
        - Extra resource: HTML at w3schools, the following chapters - HTML Basic, HTML Elements, HTML Attributes, HTML Headings, HTML Paragraphs. See also the html validator
        - lab06 LAB exercise released and due Monday by 11:59pm EDT.
- 5/30/2017
    - Lecture - Introduction to HTML. The basic idea of what an html document is, and how browsing the web works. Our first html file, and our first html lab assignment.
        - Extra resource: HTML at w3schools and see also resources linked from textbook section of the syllabus.
- 5/28/2017
    - 1pm - group chat on yuja through blackboard.
- 5/27/2017
    - A good linux tutorial - http://www.ee.surrey.ac.uk/Teaching/Unix/

    - 10am - group chat on yuja through blackboard. Note - required to do at least one per week. I am logged in now, people can join any time before 10:30am. If people are on at 10:30am I'll finish answering any questions, and then will logout.
    - The first long quiz isn't ready yet. It will be released tonight and due you'll have at least 48 hours to finish.
    - hw1 has been created, due May 31 by 11:59pm EDT. That is Wednesday night. See link at the top of this google doc to HW assignments.
- 5/26/2017
    - Lecture - using labcheck. Using the labcheck program to check your lab assignments.
- 5/25/2017
    - lab03 and lab04 assignments have been created/released. Both due May 28 by noon EDT.
    - Lecture - text editor basics - pico, vim, emacs. The very basics of three different text editors that can be used in the terminal. Make sure to take the short quiz for this video.
        - Extra resource: http://www.reallylinux.com/docs/editors/editor.shtml
    - Lecture - Using the unix shell/terminal part 1 and Lecture - Using the unix shell/terminal part 2. Unix commands, unix files, and file permissions. Make sure to take the short quizzes for these videos.
        - Extra resources: http://www.ee.surrey.ac.uk/Teaching/Unix/ http://www.reallylinux.com/
- 5/23/2017
    - Lecture "Connecting to CS server - windows", "Connecting to CS server - Mac", "Connection to CS server - linux" - watch the one for your operating system. This shows how to connect to the CS server. You can get your username and password by looking at your "My Grades" in blackboard for this course. See lab02.txt for the lab assignment for these lectures. There is no short-quiz for these.
    - **Lecture "Syllabus and Information"** - what are the goals of the course, what are the expectations of students, introducing the first lecture short-quiz and lecture lab. See lab01.txt for the lab assignment for this video. There is a short-quiz in blackboard for the video.
    - First "day" of class since I was a bit behind schedule getting everything created.
    - **Live-talk times (EST):** Sundays 2-2:30pm, Wednesdays 8-8:30pm, Saturdays 10-10:30am. We'll try it this week and see ...
    - Your responsibilities
        - **Lecture videos:** watch the video, pause and replay as needed.
        - **Lecture short-quiz:** each video will have one short quiz associated with it. The short quiz contains two questions: (1) did you watch the video? (answer truthfully, you only get credit if you watched). (2) answer a question based on something in the video. The short-quiz is available

        before you watch the video, so you can write down the question and pay attention to the video for the answer.
        - **Lab exercises:** each lecture video will have one short lab exercise associated with it. Example: create a page with a table that is 2 rows and 3 columns, red background.
        - **Weekly discussion:** a discussion thread is created each week in blackboard, and you are required to participate in the discussion by posting a question or answer at least 3 times per week. Discussion participation will be graded out of 3, with full credit given for participating at least 3 times.
        - **Weekly live-talk:** each week there are a few sessions to choose from where you will login to an online system (Yuja) and participate in a live discussion Q&A.
        - **Weekly long-quiz:** each week we will have a quiz over the most important content from the previous week.
        - **Weekly HW:** each week we will have an assignment based on the material of the previous week. Example: design your own homepage.
        - **Exams:** we will have 3, the dates are listed in the syllabus below.
        - **Project:** time permitting each student will complete a web programming project.
    - **Timing of tasks …**
        - Videos: watch videos as they appear in the youtube playlist. Videos will appear most weekdays.
        - Lecture short-quiz: lecture short quizzes will be created when lectures are made, and will be available for 48 hours after the lecture is created.
        - Lab exercises: lab exercises will be created when lectures are made, and will be available for 48 hours after the lecture is created. Note: lab exercises may require some time for you to complete.
        - Weekly long-quiz: released by 12:01am EST Monday each week, with 48 hours to complete. Weekly long-quizzes will be timed, can only be taken once, and you will not know the results until after the 48 hour window.
        - Weekly HW: released by 12:01am EST Saturday each week and due the following Wednesday 11:59pm EST.
- 5/22/2017
    - Advice on courses, programming, etc. - see everything linked off of http://cs.indstate.edu
    - Document created, woo hoo. Copy/pasted from the Steve Baker’s 151 and my normal course notes format.
    - Note - anything after here is “the syllabus” and probably won’t change once the semester starts. Anything before here is the notes from class, assignments, etc.

Table of Contents

CS 151 Introduction to Computer Science

Summer 2017 Syllabus and Information

General Information

Contact Your Instructor

Name: Jeff Kinne
Email: jkinne@cs.indstate.edu
Phone: 812-237-2136
Office: Root Hall, room A-120

Lecture, Exam, Office Hours

Lecture: all online. A normal course would have 4.5 hours per week for the summer. We will likely have 2-2.5 hours per week of lecture videos.

Exams: for each exam you have a 48 hour period in which to take the exam, from 12:01am of day 1 to 11:59pm of day two. The exams will be on: June 18/19, July 21/22, Aug 4/5. Note that the last day to drop with no grade is May 26, last days to drop with 100/75/50/25% refund are May 26, May 31, June 4, June 9.

Instructor Office Hours: I am available by email almost always. I will hold “virtual office hours” at the following times: TBD based on feedback from students.

Unix lab: The CS unix lab, room A-015 in the basement of Root Hall, will be normally open TBD hours during the summer. If you are in Terre Haute and want to work on the CS lab computers, you can come any time the lab is open. Note that during the summer there is not a lab assistant on duty to help.

Website: this google doc, or find a link from kinnejeff.com

Prerequisites

None.

Role of this Course

First course in the CS major, required of IT majors, required of a few other majors, and oftentimes the one CS course people take if they're interested in CS a little.

Recommended text

Most information required for the course is available on-line. Sources that can be used for the course include the following.

- Installing putty: http://cs.indstate.edu/FAQ/PuTTY/
- See the Using Linux section at: http://cs.indstate.edu/info/getting-started.html
- JavaScript tutorials http://www.tutorialspoint.com/javascript/
- "JavaScript: The Definitive Guide" by O'Reilly (ISBN: 978-0-596-80552-4)
- The HTML5 standard: http://www.w3.org/TR/html5/
- W3 Schools http://www.w3schools.com/
- HTML dog http://www.htmldog.com/
- CSS Basics http://www.cssbasics.com/
- Quack it http://www.quackit.com/
- Blockly games https://blockly-games.appspot.com/
- Node.js: www.nodejs.org

Course Announcements

Announcements regarding the course will be made both during class and via email to your @sycamores.indstate.edu email address. You should regularly check this email account or have it forwarded to an account that you check regularly. You can set the account to forward by logging into your indstate.edu email from Internet Explorer (the "light" version of the webmail client that opens up from Firefox or Chrome does not give the option to forward email).

Classroom conduct

You may not use cell phones, iPods/music players, etc. during class. You should be civil and respectful to both the instructor and your classmates, and you should arrive to class a few minutes before the scheduled lecture so you are ready for lecture to begin on time. You may use your computer during class if you are using it to follow along with the examples that are being discussed. You may not check email, facebook, work on other courses, etc. during class.

Course Description

The catalog description for this course is:

"History of computers and computer science, principles of process description, and problem analysis. The basic structures of sequence, iteration, and selection. Programming style, artificial intelligence, current applications."

The course really is an introduction to programming and introduction to the way many of the ISU CS courses are run (i.e., unix and programming). We have chosen to use html/css/javascript as our intro language (for this course only). Why? It is visual, and it is useful.

In terms of depth, the course approximately 1/4 HTML/CSS web design and 3/4 JavaScript programming in the web environment.

Course Outline

First Unit

- Unix, files and editing.
- Basic HTML.
- More HTML and introduction to CSS.
- More CSS and an Introduction to JavaScript programming.
- Event-Driven programming and event handlers.
- June 18/19 -   Exam #1

Second Unit

- JavaScript Variables, Data-Types and simple arithmetic operations.
- The Canvas and Numeric Bases (Decimal/Binary/Octal and Hex)
- Functions and Variable Scope. The event object. Simple Boolean logic, the comparison operators and truth tables. Introduction to conditional statements.
- More conditional statements. Introduction to loops.
- Loops (while, do-while)
- July 21/22 - Exam #2

Third Unit

- More loops (for, for-in)
- Strings and Arrays.
- Objects.
- Multidimensional arrays.
- Nested loops.
- Sorting / Computational complexity
- Aug 3/4 - Exam #3

Note - all exams are cumulative.

Course goals: you should be able to do any kind of basic web page layout/design, and also any kind of basic program. Examples: a webpage to do tic-tac-toe, calculator, collect information from a user and check it, factor numbers, sort numbers, simple space-adventure game, etc. And also, familiar and can use unix. And also, some basic CS concepts.

Grading and Assignments

The students of this course have the following responsibilities: read assigned readings before lecture, attend lecture, complete homework assignments, take in-class quizzes, take exams, and complete a project. The final grade consists of:

- **Project: 15%** of the final grade.

- **Homeworks and Quizzes: 30% total**. Most weeks there will be at least one homework assignment or quiz.

- **Exams: 45% total**. There will be 3 exams. The total exam grade will be calculated as
  max((.1 * exam1 + .15 * exam2 + .2 * exam3) / .45,
  (.15 * exam2 + .2 * exam3) / .35,
  exam3)

- **Class Participation: 10% total**. The participation grade will include (1) a component for the required interactions online, and (2) a component determined at the end of the semester based on your attentiveness throughout the semester.

CS Course Policies

Note that this course follows all standard CS course policies. In particular, (a) cheating/plagiarism by graduate students (for courses with graduate students) results in an F in the course, for undergraduates a second offense of cheating/plagiarism results in an F in the course, (b) missing 20% of the classes results in an F for any student, and (c) there will be no makeup exams. See http://cs.indstate.edu/info/policies.html for details.

Late Homeworks

All homework assignments will be given a preferred due date. Assignments can be turned in past the preferred due date, but any assignments turned in late will have their value multiplied by 50% (so the highest grade you can get on a late assignment is 50%). Each assignment will have a “final due date” past which no credit will be given.

Start Homeworks Early

I suggest attempting a homework assignment the day it is given, or the day after, so that if you have a problem you can ask early and so that if you continue to have problems in trying to complete the assignment, you will have time to ask again. Many of the homework assignments require thought and problem solving, which takes “time on the calendar” not just “time on the clock”. By that I mean that spending an hour on 3 consecutive days is likely to be more productive than trying to spend 3 hours at once on the assignment.

Expected Amount of Work

My expectation is that an average student will spend about 4-6 hours OUTSIDE of class each week (that is in addition to class time) WORKING PRODUCTIVELY/EFFICIENTLY (not just staring at the computer) to complete their coursework for this class. Some students may spend less time than this, and some students will spend more.

Note - this is your most important class, by far (for CS majors). Also, your classes should be more important than your part-time job.

Grade Cutoffs

I will design homework assignments and exams so that a standard cutoff for grades will be close to what you deserve. After the first exam I will create a grade in Blackboard called “Letter Grade” that is what your letter grade would be if the semester ended today. Initially, I will assign the following grades: 93-100 A, 90-93 A-, 87-90 B+, 83-87 B, 80-83 B-, 77-80 C+, 73-77 C, 70-73 C-, 67-70 D+, 63-67 D, 60-63 D-, 0-60 F

My goal is that the different grades have the following rough meaning.

A+/A
You understand everything and probably could teach the course yourself.

B+/A-
You understand nearly everything, and should be all set to use this knowledge in other courses or in a job.

C/C+/B-/B
Some things you understand very well and others you don’t (more towards the former for a B and more towards the latter for a C).

D-/D+/C-
You did put some effort in, and understand many things at a high level, but you haven’t mastered the details well enough to be able to use this knowledge in the future.

F
Normally, students that get an F simply stopped doing the required work at some point.

Blackboard

The course has a blackboard site. Click here to go to blackboard. You should see this course listed under your courses for the current term. The blackboard site is used for (a) giving you your grades (go to the course in blackboard, then click “My Tools”, and then “My Grades”), and also (b) some required online interactions (e.g., discussion of assignments). All course content, schedule, etc. is kept in this google doc (which you are currently viewing). Video lectures for the course are kept in a youtube playlist linked from the top of this document.

Academic Integrity

Please follow these guidelines to avoid problems with academic misconduct in this course:

- **Homeworks:** You may discuss the homework assignments, but should solve and finish them on your own. To make sure you are not violating this, if you discuss with someone, you should DESTROY any work or evidence of the discussion, go your separate ways, SPEND at least an hour doing something completely unrelated to the assignment, and then you should be able to RECREATE the program/solution on your own, then turn that in. If you cannot recreate the solution on your own, then it is not your work, and you should not turn it in.

- **Note on sources:** if you use some other source, the web or whatever, you better cite it! Not doing so is plagiarism.

- **Exams and quizzes:** This should be clear - no cheating during exams. The exams will be closed-book, closed-notes, no computer, and no calculator.

- **Projects:** You should not copy from the internet or anywhere else. The project should be your own work. It will be fairly obvious to me if you do copy code from the internet, and the consequences will be at the least a 0 on the project.

If cheating is observed, you will at the least receive a 0 for the assignment (and may receive an F for the course), and I will file a Notification of Academic Integrity Violation Report with Student Judicial Programs, as required by the university's policy on Academic Integrity. A student who is caught cheating twice (whether in a single course or different courses) is likely to be brought before the All-University Court hearing panel, which can impose sanctions up to and including suspension/expulsion. See the Student Code of Conduct and Academic Integrity Resources for more information.

Please ask the instructor if you have doubts about what is considered cheating in this course.

Special Needs

If you have special needs for the classroom environment, homeworks, or quizzes, please inform the instructor during the first week of classes. If you have any such needs, you should go to the Student Academic Services Center to coordinate this. See Student Academic Services Center - Disabled Student Services for more information.

Disclosures Regarding Sexual Misconduct

Indiana State University fosters a campus free of sexual misconduct including sexual harassment, sexual violence, intimate partner violence, and stalking and/or any form of sex or gender discrimination. If you disclose a potential violation of the sexual misconduct policy I will need to notify the Title IX Coordinator. Students who have experienced sexual misconduct are encouraged to contact confidential resources listed below. To make a report or the Title IX Coordinator, visit the Equal Opportunity and Title IX website: http://www.indstate.edu/equalopportunity-titleix/titleix.

The ISU Student Counseling Center – HMSU 7th Floor | 812-237-3939 | www.indstate.edu/cns
The ISU Victim Advocate – Trista Gibbons, trista.gibbons@indstate.edu
HMSU 7th Floor | 812-237-3939 (office) | 812-230-3803 (cell)
Campus Ministries - United Campus Ministries | 812-232-0186
http://www2.indstate.edu/sao/campusinistries.htm
www.unitedcampusministries.org | ucmminister2@gmail.com
321 N 7th St., Terre Haute, IN 47807

For more information on your rights and available resources
http://www.indstate.edu/equalopportunity-titleix/titleix