Week 1 Study Guide

- What is an algorithm? Provide some examples, both in the context in of computer science and beyond.
  - An algorithm is a set of instructions to solve a problem.
  - An algorithm is language-independent and there are many possible algorithms for the same problem.
  - An example of an algorithm is how to cut the grass or do dishes. In relation to computer science, an example is the “Wall Follower”.
- What is a heuristic? How do they relate to algorithms?
  - A heuristic is additional logic or “hints”.
  - The term heuristic is used for algorithms which find solutions among all possible ones ,but they do not guarantee that the best will be found, therefore they may be considered as approximately and not accurate algorithms.These algorithms usually find a solution close to the best one and they find it fast and easily. Sometimes these algorithms can be accurate, that is they actually find the best solution, but the algorithm is still called heuristic until this best solution is proven to be the best.
- What is the brute force method? What are some problems with it?
  - In computer science, brute-force search or exhaustive search, also known as generate and test, is a very general problem-solving technique and algorithmic paradigm that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement.
  - The main disadvantage of the brute-force method is that, for many real-world problems, the number of natural candidates is prohibitively large. For instance, if

we look for the divisors of a number as described above, the number of candidates tested will be the given number n.

- What is version control? How is it different than simply *saving* a file?
  - Version control is the practice of tracking changes to files and being able to look at or revert to older revisions.
  - It is different from simply saving a file as it also allows easier collaboration and doesn't result in the need to save a file for each revision.
- What's the difference between Git and Github?
  - Git is a program, where as Github is a company/ website. Github is one application in which we are able to push files to a remote repository.
- Git commands to know:
  - git log
  - git init
  - git status
  - git add
  - git commit with -a and -m
  - git remote add
  - git remote -v
  - git push
- Why is using the command line important?
  - Using the command line is important because it is much faster and easier than working with a GUI, along with allowing you to do many more actions. With the command line, you can do powerful things succinctly
- What is the prompt in the context of command line?
  - If you see the prompt, it means the shell is waiting for you to give it a command.
- Shell commands to know:
  - ls - list contents of directory
  - pwd - print working directory

- cp - Copy
- mv - Move and Rename
- cd - Change directory
- mkdir - Make Directory
- rm - Remove File(s)
- man - Print command details
- any command with --help
- CTRL-D and CTRL-C
    - Ctrl-C terminates the currently running command or a process. Ctrl-D is an equivalent of exit command. You can think about the Python REPL, where Ctrl-C ends the currently running command, but Ctrl-D exits the REPL itself

- What's the difference between relative and absolute URLs?
    - An absolute path refers to the same location in a file system relative to the root directory (/home/hackbright/src), whereas a relative path points to a specific location in a file system relative to the current directory you are working on (src).
- What's the difference between parameters and arguments?
    - The list of things defined for a function to receive. When you create a function with def calculate_order_total(price, qty), we'd say that this has two parameters, price and qty.
    - When you call a function, you provide arguments to it—so calling that function, like calculate_order_total(9.99, 10) has two arguments, 9.99 and 10.
- What's the difference between return, break, and print?
    - Return: Exits function and hands value back to call (but doesn't print)
    - Print: Prints on screen but doesn't exit function
    - Break: Terminates the current loop and resumes execution at the next statement

- What does a function return if it doesn't have a return statement?
  - A function will return None if it does not have a return statement.
- What is a default parameter?
  - A default parameter is a value that will be passed into the function as a default argument if one is not specified when the function is called.
- What is scope? What is function scope?
  - Scope is where a variable name is meaningful.
  - Function scope refers to the fact that variables are only usable within the function, but not outside of it. They would have to be in the global scope to be used in any and all functions.
- What are some important Python style considerations for writing functions?
  - Functions start with the term **def** and is followed by the function name which is best to be written in snake_case. Function names can also not start with a number, but instead a letter or underscore. The function name is followed by parenthesis where the function parameters are listed. You must end the function with a colon (:) or you will get an error.
  - When writing functions, you should always be mindful of indentation as it helps organize the code and you will also receive errors if it is not correctly indented.
- Python list methods/functions to know:
  - sorted() - sorts list into new list (descending by default)
  - .sort() - sorts and modifies the actual list
  - .append() - add a single element to the end of collection
  - .extend() - adds a collection to the end of another collection
  - How to index a list
    - list[index] —> grades[0]
  - How to slice a list

        - list[start:stop:step]; step is optional
    - How to loop over a list
    - Using range(len(some_list))
    - Using a counter and indexing the list
    - Without using either of the above (just a for-loop)
- What is mutability?
    - Mutability refers to the ability to mutate/modify a collection type. For example, lists are mutable, meaning you can reassign the value of an item contained within it.
- How is Python memory different than C memory? How are Python variables different than “classic” variables?
    - Python memory is different than C memory in the fact that a new space of memory is created and pointed to when variables are reassigned, where as in C, the memory space of the original variable is just modified to accommodate the new value.
    - Python doesn’t have variables (a fixed place in memory) in the classic sense, it instead has identifiers —a name that points to a spot in memory that can move.
- What is garbage collection?
    - Garbage collection refers to the clearing of unused memory spaces that occurs at the end of a program and also unknowingly during the program.
- What is the id function? What does the comparator is do in Python?
    - The id function returns the id of an object.
    - The comparator “is” checks to see if two objects have the same id/are the same object and returns True or False.
- What are sets good for?
    - Sets are good for unique, unordered collections of items.

- Set methods to know:
  - .add() - Insert item into the set
  - len() - Get the length of the set
  - .remove() - Remove a specific item from the set
  - How to create a set
    - moods = set(['happy', 'sad', ‘cheerful’])
    - moods = {'happy', 'sad', ‘cheerful’}
    - moods = ['happy', 'sad', 'sad', 'cheerful', 'happy', ‘sad’]; moods = **set**(moods)
  - How to check if something is in a set
    - “happy” **in** moods
- What are tuples good for?
  - Tuples are good for groupings of data with meaningful order
- Tuple methods to know:
  - How do you create a tuple?
    - fruit = ('ripe', 'Red Delicious')
  - How to index a tuple
    - fruit[0]