    - investigating/fixing a bug
    - organizing and making sense of data
    - making something efficient
    - designing code to be generic and reusable
    - making something really useful that users enjoy
- There are several styles of collaborating with other people. For example:
    - Having a design team spec everything out in detail, then working on your own from the spec, integrating at defined checkpoints
    - Having periodic (once every two weeks, or weekly) meetings to figure out the design and interfaces together, then working separately and integrating later
    - Have frequent (e.g. daily) meetings to coordinate with others, often working in a lab where everyone sits together and can overhear each other's conversations
    - Sitting side by side and programming together (pair programming)

  Which do you prefer and why?
- What do you find to be the most frustrating aspect of working on a software project?
- What do you consider "done"?

Culture Fit Questions

- How do you ensure you are delivering what was asked for?
- How do you ensure code quality?
- How do you learn best? For example:
    - Have a teacher present a topic, then try it yourself
    - Jump in and try it
    - Read manuals/documentation first, then try it yourself
    - Work with a peer who is teaching while you are trying something together
- If you were going to start a new company and build a new product from scratch, how would you go about it?
- Describe a time when you worked with another person on a project who did not complete their work on time or with sufficient quality. How did you handle this situation?
- What would your co-workers or fellow students say are the best things about working with you? ...and the worst?
- What is work environment works best for you? (e.g. working in teams, working alone, being in a very formal environment with clear processes and procedures or a more loosely defined environment where it is up to you to ask questions if you want clarity or detail?)
- What is the worst behavior you've seen from a former manager (or teacher if no industry experience): what happened and what did you do about it?
- What do you like most about your current job that you hope you'll find in a new job?
- What aspect of your current job would you most like to change?
- What motivates or excites you about work?
- Describe a time when a project (work or school) was late or did not perform as expected. What did you do? What did you say to your boss/teacher?
- Describe a time when you went out of your way to make a customer happy. What happened, what did you do, and what was the customer response?
- What's the most recent thing you learned on the job? What's something you want to learn?
- How do you stay up to date and keep your skills sharp?
- What's a project you've worked on that you're most proud of?
- Describe the work environment in which you are most productive and happy.
- What are some things you look for when reviewing somebody else's code?
- What's one thing you like about your current/prior job that you'd want here as well?
    - What's one thing you dislike that you'd like to leave behind?
- What's something your current/previous coworkers would say they liked about working with you?
    - What's something they would say you need to improve on?

Technical Questions

Java

- Name as many primitive data types in Java that you can recall

  EXPECTED: some of: byte, char, short, int, long, float, double, boolean, but NOT string
- What is the difference between and interface and an abstract class?
  EXPECTED: an interface defines method signatures and defines un-modifiable variables, abstract classes can define default

method implementations, but cannot be instantiated

- What is multiple inheritance, and does Java support it? Why or why not?
  EXPECTED: Inheritance is when one class extends another, and gets all of the methods and variables of the parent class. Java does not support multiple inheritance, using single inheritance and interfaces instead, because multiple inheritance gets messy and complicated (see: C++)
- What is synchronization in Java (or what is the 'synchronized' keyword in Java) – what does it do any why would you use it
  EXPECTED: synchronization is a method within Java to control access to methods or blocks of code by different threads – it is used to prevent multiple threads from accessing the same resources at the same time (potentially corrupting data)

Python

- Name as many primitive data types in Python that you can recall

  EXPECTED: some of: integer, float, complex, str, boolean, none. May also include tuple, list, set, dictionary
- What is the difference between a deep copy and a shallow copy?
  EXPECTED: A shallow copy retains the references to the original object, so that changing the original object also changes the copy. A deep copy creates a brand new object and makes a copy of all of the data in the original object, so that both the original and the copy are independant
- What is pickling and unpickling in Python?
  EXPECTED: pickling is the process of writing an object to a file using a dump function, and unpickling is the process of restoring that object from a file

Perl

- How are primitive data types handled in perl (e.g. what is the difference between a variable holding a string vs a number, vs a reference?)

  EXPECTED: something along the lines of: perl is not a strongly typed language. Data types are converted dynamically based on context. [bonus: everything is a byte array behind the scenes]
- What does the chomp() function do in Perl. What happens when you pass an array to the chomp() function?
  EXPECTED: chomp() trims any newlines, carriage returns, spaces, etc. off the end of string. If called on an array, it chomps all of the values in the array
- How do you pass variables to function in perl? Are values passed by reference or by value?
  EXPECTED: values are passed by value to perl functions, and are accessed by the default array variable name '@_' inside the function. It is possible to pass the value of a reference which approximates pass by reference

SQL

- What is a primary key? What is it used for?
  EXPECTED: a primary key is used to define how rows in a database table should be identified to make them unique. Defining a primary key usually also implies an index is created to find rows more quickly.
- Can you have a table without a primary key? If so, why might you want to do this?
  EXPECTED: YES, if you need to be able to allow duplicate rows in a table, you might consider creating it without a primary key
- What is the difference between and inner and an outer join?
  EXPECTED: With an inner join, ever row in one table must match a corresponding row in a second table. Within an outer join, all matching rows in one table are return whether or not there are corresponding rows found in the second table
- What are somethings to look for when a query is slow. What things can be done to make queries faster?
  EXPECTED: some discussion of indexing, using explain plan to see if indexes are being used, maybe re-ordering queries or partitioning data, or defining foreign keys for more efficient joining, etc.
- What is a partition and why you might use them?
  EXPECTED: Breaking the data in an application's database into separate, smaller pieces, or partitions. This can improve scalability and performance.
- What are the different types of JOINS?
  - INNER JOIN: With an inner join, every row in one table must match a corresponding row in a second table
  - OUTER JOIN
    - FULL OUTER JOIN: produces the set of all records from the left table and the right table, with matching records from both sides where available. If there is no match, the missing side will contain null.
    - LEFT OUTER JOIN: You get all the rows from the left table, plus the matching rows from the right table (where available). If there is no match, the right side will contain null.
    - RIGHT OUTER JOIN - You get all the rows from the right table, plus the matching rows from the left table (where available). If there is no match, the left side will contain null.
  - CROSS JOIN (cartesian product): Joins every row in left table to every row in the right table
  - NATURAL JOIN: joins two tables based on same column attribute name and datatypes. The resulting table will contain all the attributes of both the table but keep only one copy of each common column.

- What are indexes used for?
  EXPECTED: Indexes improves the speed of data retrieval operations on a database table at the cost of additional writes and storage space to maintain the index data structure.
- What can you tell me about these types of indexes?
  - Unique index
  - Clustered index
    - A clustered index defines the order in which data is physically stored in a table.
    - Clustering indexes can improve the performance of most query operations because they provide a more linear access path to data, which is stored in pages. In addition, because rows with similar index key values are stored together, sequential detection prefetching is more efficient when clustering indexes are used.
  - Covering index
    - Covering index contains the data to be searched through within the index, so that the SQL query can get the required data without reading from the table data.
      CREATE INDEX person_id_fname_lname ON person (person_id) INCLUDE (fname, lname);
      SELECT fname, lname FROM person WHERE person_id = 123;
    - All data can be directly read from the index without going to the table to get the additional columns
    - Not including the additional columns in the actual index can reduce insert times because the additional data doesn't need to be sorted.

Data (storage and data structures)

- What is the difference between and array and a linked list? What are some pros and cons of each?
  EXPECTED: an array is typically fixed length (note: some languages call structures arrays which aren't implemented as arrays under the hood) which each item the array being assigned a position (element 1, 2, 3, ... n). A linked list is a structure where each element in the list is connected to it predecessor (single-linked list) and possible also its successor (double-linked list). Array are very efficient for random access, but are difficult to grow/shrink in size. Linked lists are very efficient for growing/shrinking, but are slow for random access
- What is a hash table or dictionary, and how does it work?
  EXPECTED: a hash table works by computing a fixed-length number (a.k.a. has value) for each value that needs to be stored, and then placing each value into a predefined "bucket" corresponding to the calculated number. It would be similar to allocating a large array, and then using the computed hash value as the index to store the value in that array. It allows for very fast insertion and fast random access, at the expense of space. It is often used for relational data, in the form of storing and manipulating key-value pairs.
- When might using a hash table or dictionary be a poor design choice?
  EXPECTED: this is somewhat open-ended, but the answer should mention some of the trade-offs of using hashes, namely memory space, or where the data would involved a large number of hash collisions, or when the data does not lend itself to a key-value format.
- What is a queue? Describe how you might used a linked list to implement a FIFO queue. What about a FILO queue?
  EXPECTED: a queue is a list of data waiting to be processed. In a FIFO queue, items would be added on one end of the list, and items removed from the other end. In a FILO queue, items are added and removed from the same end of the list
- Can you describe a data storage scheme other than a traditional relational database, and how it works (e.g. NOT Oracle, MySQL, PostgreSQL, MSSQL, etc.)
  EXPECTED: any discussion of a non-relational database, such as:
  - a document model (like elastic search, with a bunch of individual documents along with indexes of attributes of those documents),
  - a graph database, where data is broken down into key-value pairs and keys the relationships between all of the keys are defined so that each element can be related to at least one other element
  - a "NOSQL" data store like MongoDB or Redis where data is broken into key-value pairs and indexed (either in-memory or on disk) and values are either unstructured data, or formatted data (such as JSON)
  - a HDFS or some other distributed file system where data is broken up into key-value pairs and distributed across a large number of servers

Data Processing

- Given a single 100GB file of English text, how might you find the top 10 most frequently used words?
  EXPECTED: solution need to address the size of the data, and how much memory would be used. Probably a hash would be useful.
- Imagine a data set where each row of a file has an unique id number, but where the data set is too large to fit into memory. How would you sort all of the rows in the data set?
  EXPECTED: solution should involve some sort of divide and conquer strategy -- sort subsets of the data set and glue them back together later. Bonus points if they mention merge sort
- Imagine a data set containing 100 million emails spread across 1000's of files. How would you find all of the unique FROM: addresses in that data set?
  EXPECTED: solution should recognize that the data would not fit into memory. The solution should also be able to process data in parallel, and probably be multi-stage (step 1, extract emails, step 2, reduce the data set by removing duplicates, etc.)

Object Orientated Programming

- What is encapsulation, and why is it important
  EXPECTED: encapsulation is the idea of hiding implementation details of a function or system of objects so that the code, logic, storage mechanism, etc. can be changed without breaking any code that calls that function. It allows more flexibility to change the behavior of existing systems, and avoids the sort of tight-coupling of code that makes code maintenance and enhancement difficult to perform
- Describe the difference between inheritance and implementing interfaces
  EXPECTED: inheritance allows common functionality to be coded once, and re-used by other related objects, implementing an interface advertises that an object behaves in a particular way, allowing otherwise unrelated objects to be accessed and used in a common way
- What is a static method, and why would you use one?
  EXPECTED: a static method is code that only gets instantiated once, meaning there is only one copy of the code that exists in memory while the program is running. If you have a function that does not need to preserve state, then using a static method is more memory efficient, as there are fewer copies of the method in memory.
- What are some advantages of Object Oriented Programming vs. strictly procedural programming?
  EXPECTED: some discussion of code reuse, ability to refactor, encapsulation, matching the code more closely to the problem being solved, easier to multi-thread, etc.

Algorithms

- Are you familiar with algorithm analysis, time/complexity trade-offs and Big-O or O(n) notation? Can you briefly explain this?
  EXPECTED: some concept of looking at algorithms to determine how efficient they might be. Big-O / O(n) notation is away of describing (at a high level) the cost of an algorithm in terms of computing resources, so that alternative algorithms can be compared and describes how an algorithm scales with input
- Example: you see an algorithm that contains two nested loops, with a function call inside the inner loop – what things would you look at to see if this algorithm needs to be optimized?
  EXPECTED: some things that should come out of the discussion: how much data does this algorithm process? – maybe it is a tiny data set and a doubly-nested loop isn't a big problem, what does the function do – are there even more loops inside the function, or an expensive API or function call?, is it possible to unwrap the loops and process the data in a single pass?
- Can you describe an example of when trading off using more space in order to spend less time processing a data set might be a good trade off?
  EXPECTED: something around caching data, not looking up values more than once, not calculating values more than once, pre-processing the data to map related elements once (rather than having to repeatedly map relationships inside an inner loop) etc.

Open-ended Questions / Design

- You are writing a program that manages a parking garage. The system assigns spaces to drivers as they enter the garage. (Assume all drivers park in their assigned slots 100% of the time). All of the parking spaces are the same size, but two motorcycles can fit into one space.
  - What are the classes/interfaces you would start with?
  - What are the primary methods and data does each class contain?
  - What is the primary data structure inside the program and how does it work?
  - What is the primary data storage mechanism behind the system and how is it organized?

Short Programming Exercises (these are possible to do over a video call in 20-30 minutes or less)

- Write a program to read in the following list of ip addresses from a file and print them out in ascending order:

  48.97.1.9
  12.14.230.6
  48.98.1.8
  12.140.21.4
  36.5.0.10
  120.0.0.4
  48.97.1.8
  12.140.201.4
  36.5.0.1
  48.97.2.8
  120.0.1.4

  EXPECTED:

12.14.230.6
12.140.21.4
12.140.201.4
36.5.0.1
36.5.0.10
48.97.1.8
48.97.1.9
48.97.2.8
48.98.1.8
120.0.0.4
120.0.1.4

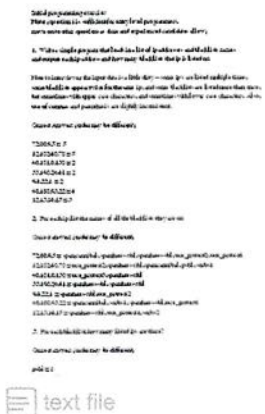
- Write a program to read in the following list of names, and count how many people there are in each family (note that all variations on Delgado/Tucker/Pritchett are ONE family for this exercise):

  Frank Underwood
  Claire Underwood
  Homer Simpson
  Marge Simpson
  Lisa Simpson
  Bart Simpson
  Maggie Simpson
  Walter White
  Skyler White
  Walter White Jr.
  Jay Pritchett
  Gloria Delgado-Pritchett
  Mitchell Pritchett
  Manny Delgado
  Lily Tucker-Pritchett

  EXPECTED:

  Underwood = 2
  Simpson = 5
  White: 3
  Prichett: 5

- VERSION 1 (easier): Given the following data set, write a program to answer the following questions (note, this can be time boxed by asking all the questions, or only 1 or 2):
  - Note to interviewer: the input data is a little dirty -- some ip's are listed multiple times, some blacklists appear twice for the same ip, and some blacklists are listed more than once, but sometimes with upper case characters, and sometimes with lower case characters.
  - Files for the exercise above:

Data File	Questions and Answers	Example Solution (in perl)
data.txt	 text file	example_solution.pl

12.13.14.15=spamhaus-xbl
12.13.14.15=sorbs2
68.180.93.22=spamcannibal
12.102.40.70=uce_protect2
72.88.9.5=uce_protect1
68.180.93.22=SORBS1
72.88.9.5=spamhaus-xbl
33.190.26.41=spamhaus-xbl
12.102.40.70=sorbs1
68.101.8.130=uce_protect3
72.88.9.5=uce_protect2
12.13.14.15=uce_protect1
72.88.9.5=spamcannibal
68.180.93.22=spamhaus-sbl
12.102.40.70=Spamhaus-xbl
68.180.93.22=sorbs1
72.88.9.5=spamhaus-sbl
9.4.22.1=spamhaus-xbl
12.102.40.70=spamcannibal
9.4.22.1=uce_protect2
68.180.93.22=uce_protect1
12.102.40.70=psbl
68.101.8.130=spamhaus-xbl
33.190.26.41=spamhaus-sbl
9.4.22.1=uce_protect2

- Write a simple program that loads in a list of ip addresses and blacklists names and outputs each ip address and how many blacklists that ip is listed on.
  EXPECTED (might be in any order)

  72.88.9.5 = 5
  12.102.40.70 = 5
  68.101.8.130 = 2
  33.190.26.41 = 2
  9.4.22.1 = 2
  68.180.93.22 = 4
  12.13.14.15 = 3

- For each ip, list the names of all the blacklists they are on.
  EXPECTED: (might be in any order)

  72.88.9.5 = spamcannibal, spamhaus-xbl, spamhaus-sbl, uce_protect2, uce_protect1
  12.102.40.70 = uce_protect2, spamhaus-xbl, spamcannibal, psbl, sorbs1
  68.101.8.130 = uce_protect3, spamhaus-xbl
  33.190.26.41 = spamhaus-sbl, spamhaus-xbl
  9.4.22.1 = spamhaus-xbl, uce_protect2
  68.180.93.22 = spamcannibal, sorbs1, spamhaus-sbl, uce_protect1
  12.13.14.15 = spamhaus-xbl, uce_protect1, sorbs2
- For each blacklist, how many listed ips are there?
  EXPECTED: (might be in any order)

  psbl = 1
  sorbs1 = 2
  spamhaus-sbl = 3
  uce_protect1 = 3
  uce_protect2 = 3
  sorbs2 = 1
  uce_protect3 = 1
  spamcannibal = 3
  spamhaus-xbl = 6

- VERSION 2 (harder): Given the following data set, write a program to answer the following questions (note, this can be time boxed by asking all the questions, or only 1 or 2):

- Note to interviewer: the input data is a little dirty -- some ip's are listed multiple times, some blacklists appear twice for the same ip, and some blacklists are listed more than once, but sometimes with upper case characters, and sometimes with lower case characters.
- Files for the exercise above:

Data File	Questions and Answers	Example Solution (in perl)
text file	text file	example_solution.pl

SQL technical exercises

There are two versions of the SQL exercises. They both use https://www.db-fiddle.com/ so that candidates don't have to have a running copy of a database server to code against.

The "screen" version of the exercise is designed to be used during a phone screen to quickly assess a candidates SQL skills. The questions are designed to be easy and get slightly harder, but are not exhaustive.

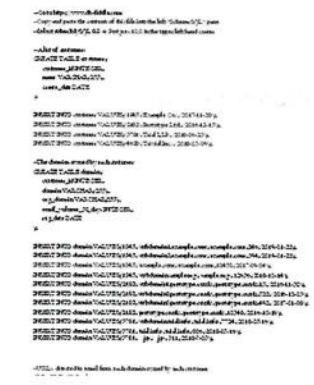
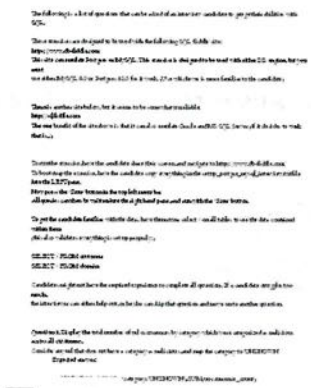
The "interview" version of the exercise is harder and parts of it are more open-ended to trigger more critical thinking.

Each exercise as two files: a setup file which should be sent to candidates a few minutes before the interview starts, and a "Q_and_A" file which includes questions and example responses.

"Screen" version for doing phone screens

Setup File	Questions and Answers
text file	text file

"Interview" version for doing on-site interviews

Setup File	Questions and Answers
 text file	 text file

SQL Exercises - Queries

Below is some code that can be used to test the SQL level of a candidate. These queries start off easy and progress to more advanced.

- The first block contains the DDL needed to build and populate the tables
- The second block of code contains some possible solutions to the queries.

I have been setting the tables and data before the interview using DB Fiddle, then I can save a link to the page and just send it via chat during the interview.

For example: https://www.db-fiddle.com/f/7Fqc9jDKwW4L8MN9J6jx41/6

DDL For SQL Exercises Expand source

SQL Exercise Solutions Expand source

DevOps Technical Exercises

1 hour Technical Screen:

1. Build a docker image
2. Run image
3. Tag Image
4. Modify code
5. Update image
6. Run image
7. Change port
8. Run old tagged image (simultaneously)

Python Docker Image

python code

setup 2 images on different ports

Onsite Technical Screen:

Web Technical Exercises

Dog API

- Use Case: **first technical screen** – tests basic JavaScript concepts like fetching from API, parsing JSON responses, basic React-isms
- Boilerplate (React 17)
  - **Functional / With Hooks (preferred):**
    - https://stackblitz.com/edit/react-bvzgfx?file=src/DogApp.js
    - https://codesandbox.io/s/dog-api-boilerplate-rytws
  - Note: CodeSandbox has recently been crashing some interviewees' browsers
- Solved
  - React 17 - Solution #1: https://stackblitz.com/edit/react-huyukk?file=src/DogApp.js
  - React 17 - Solution #2: https://stackblitz.com/edit/react-bco9yy?file=src/DogApp.js
  - React (Lifecycle): https://codesandbox.io/s/qz6pzz39m9
  - Angular: https://codesandbox.io/s/jkd0jkdfjkd-m98sb
- Expectations
  - Candidate should be able to quickly fetch data and place it in the correct hook (**useEffect**)
  - Candidate should know which hook to use to store their data in for accessing later (**useState**)
  - Candidate should know what JS functions to use to iterate over an Object
    - Object.entries() or
    - for...in... syntax
  - Candidate should know how to map over data in JSX to display HTML elements dynamically
  - Candidate should be able to display list of dogs & random image on click in allotted time (~35min)
    - If they complete both parts, advance
    - If they don't, generally reject
- Common hangups
  - If interviewee forgets to include an empty dependency array [] in the **useEffect**, they'll trigger an infinite loop
  - If interviewee wants to use Axios, they MUST use Axios prior to version 1.0, as anything later than 1.0 doesn't work with Stackblitz.
- Outdated Boilerplate
  - Class / Lifecycle Methods (this is pretty "React 2018" so try to avoid this)
    - https://stackblitz.com/edit/react-hyez9s?file=src/DogApp.js
  - Angular: we rarely administer this one...even if a candidate's expertise is in Angular, we want to see how they do in React with the aid of Google 🙂
    - https://codesandbox.io/s/angular-dog-api-boilerplate-cldkf

Random People Table

- Use case: **second technical screen** – very open-ended, tests how a participant works through a large dataset, how they perform when provided little guidance, how they design something more complex
- Boilerplate (React 17)
  - https://stackblitz.com/edit/react-aytskx?file=src%2FPeopleApp.js
- Solved
  - React 17 - Solution #1: https://stackblitz.com/edit/react-paqubw?file=src%2FPeopleApp.js
- Expectations
  - Candidate should be able to quickly fetch data and place it in the correct hook (**useEffect**)
    - especially since they did it in the first screen
  - Candidate should know which hook to use to store their data in for accessing later (**useState**)
    - especially since they did it in the first screen
  - Candidate should be able to quickly understand how to structure an HTML table via documentation even if they aren't familiar with it off the top of their head
  - Candidate should understand how to write **sort** function
  - Candidate should be able to display table & implement sort in allotted time (~35min)
    - If they complete display + sort, advance
    - If they don't, generally reject
- Common hangups
  - We include a hint to flatten the data structure to make sorting easier...most people miss this hint
  - if interviewee forgets to include an empty dependency array [] in the **useEffect**, they'll trigger an infinite loo

› Exercises we no longer administer

Async Fetch

https://stackblitz.com/edit/async-onclick?file=src/App.js

- Use Case:
  - Usually our 2nd technical screen
  - They did well on Dog API and shows basic understanding of fetching / parsing JSON responses. We want to gain a better understanding of their skillset from a different angle.
- Boilerplate
  - React:
    - https://stackblitz.com/edit/async-onclick-ybrfwm?file=src/Sandbox.js
    - https://codesandbox.io/s/ko6k1zy34o
      Note: CodeSandbox has recently been crashing some interviewees' browsers
  - Vanilla JS:
- Solution

Twelve Days of Christmas

- Use Case:
  - Entry-level hires who don't necessarily know JavaScript well enough to do the Dog API but have basic programming concepts down.
  - Let them use whatever language they want.
- Question:
- Solution:

No labels