1. What is your favorite programming language?

Right now java is my favorite because ive been using it the most. I love the garbage collecting

Strongly typed language, impertative, and I love how theres so many libraries that are free to use and easy to find. I really enjoy thinking about object oriented design and the exception handling java offers. After learning C in college, using java is so much easier and there is a lot less to worry about. But also i like that its easy to know exactly whats going on under the hood with java. Java was the first programming language I learned too, way back in freshman year of highschool over 10 years ago. I programmed with it in plain text, not even IDEs. We compiled our code using windows cmd javac and ran the programs with java commands.

Java also has powerful ides and junit testing.

You don’t have to worry about pointers, worrying about pointers isn’t a bad thing. Its just that its one less thing to worry about when programming. I also like how java code works everywhere.

Static typing is also nice. I don’t have to define my variable before I use it.

Those are also reason why I really want to get into learning more about python because of the ease of use, and how many more libraries there are in python!

1. What was the hardest coding assignment you’ve done.
2. Given 2 numbers, how would you swap their values without using math, built-in functions or a third variable.

A = A^B

B = A^B

A=A^B

1. findRecursiveFiles(File arr[],String filename)

{

For(file f : arr)

{

If(f.isFile())

{

If(f.getName().equals(filename)

{

System.out.println(“found!”);

}

}

Else if(f.isDirectory()

{

findRecursiveFiles(f.listFiles, filename);

}

}

}

1. How to delete all the files in a specific directory?

Use rm –rf /directory/\* the r means recursive delete and the f forces files to be deleted without making the user go through a prompt. And the directory/\* allows u to delete the files but not the directory itself

1. In UNIX, how do you list all the files in a specific directory?

Ls –(a|l) ./directory/

1. In UNIX What does Grep stand for and what does it do?

Grep stands for globally search a regular expression and print

It searchs a plain text file using a regular expression and prints the lines that match it

1. Find max value in an array without built-in functions

Int Findmax(int arr[]){

Int max = arr[0];

For(int I = 0; i<arr.length; i++)

{

If(arr[i] > max)

Max = arr[i];

}

Return max;

}

1. Given an array of integers, determine which integers are prime numbers

**public** **static** **void** findPrimes(**int** arr[])

{

**for**(**int** i =0; i< arr.length;i++)

{

**if**(*isPrime*(arr[i]))

System.***out***.println("at index "+i+" : "+ arr[i] + " is a prime.");

}

}

1. Given a text file, how do you find all the unique words inside the text file?

I would use unix built-ins to solve this problem.

Cat file.txt | tr ‘ ‘ \n | uniq –u

1. Find word with max occurances in a given text file

Cat file.txt | tr ‘ ‘ \n | uinq –c | sort –nr | head –n 1

1. How would you show the first 100 prime numbers?” (How to get all prime numbers between 1-100)

Use the isprime from up there.

**private** **static** **boolean** isPrime(**int** n)

{

**for**(**int** f = 2; f<n; f++)

{

**if**(*isFactor*(f,n))

**return** **false**;

}

**return** **true**;

}

**private** **static** **boolean** isFactor(**int** f, **int** n)

{

**return** n % f == 0;

}

1. To find duplicates.

Cat file.txt | tr ‘ ‘ \n | uniq –d

1. Given a website filled with bear facts, how would you count all the times the word 'bear' has occurred? Be aware that 'bear' in double quotes should be included in the count, but the word 'unbearable' should not be included.

I would use unix again and do cat website.html | tr ‘ ‘ [\\n](file:///\\n) | grep –w “bear” | wc –l

1. What if there were multiple websites of bear facts? How would you keep track of the count for each website?

I would store the name and count to a file

Use a bash script to do this where it would add to a file

1. What is the modulus operator?

It finder the remainder of a division operation. So 7 mod 2 would be 1.

1. Given two .txt files containing JSON strings with the same keys but different values, how would you parse the strings to determine which keys contain the same values? You cannot use any external modules

So json files the keys are always strings and the values are some type of json format type.

If we arenta using external libraries then we must read every line by line

If I knew what object they create then I would make that class for it, as the keys and values come in from the file I would set them as they come. Then compare the objects and see which keys have the same values!

1. Difference between b-tree and binary.

In a b-tree all leaves are on the same level.

A b-tree needs a minimum degree of t. it depends on disk block size usually.

Every node must contain t-1 keys

All nodes may cotain 2t-1 keys at most.

The number of children of a node is equal to the number of keys in it plus one

All keys of a node are sort in increasing order.

The b-tree grows and shrinks from the root, unlike the binary tree which grows downward or shrinks downward.

Time to search, insert,delete is o(logn)

Binary tree search insert delete is o(n)

1. Have you heard of a Directed Acyclic Graph?

with each edge directed from one vertex to another, such that there is no way to start at any vertex *v* and follow a consistently-directed sequence of edges that eventually loops back to *v* again

1. how do you find a loop in a faulty binary tree structure?

Print its inorder traversal and if there is a duplicate then there is a loop.

1. Implement breadth-first search.

Void bfs(int s)

{

Boolean visited[] = new Boolean[vertex\_size];

LinkedLIst<int> queue = new linkedlist<>()

Visited[s] = true;

Queue.add(s)

While(queue.size() != 0)

{

s.queue.dequeue();

print s

list<node> neightbors = s.getneightbors

for(int I = 0l i< neighbors.size(); i++)

{

int n = i.next();

                 if (!visited[n])

                 {

                     visited[n] = true;

                     queue.add(n);

                 }

}

}

}

1. Given a 3 gal buckeyt and 5 gal. measure out 4gal.
2. Find fib

**static** **int** fib(**int** n)

{

**if** (n <= 1)

**return** n;

**return** *fib*(n-1) + *fib*(n-2);

}

**public** **static** **int** fib3(**int** n, Map<Integer,Integer> lookup)

{

**if**(n<= 1)

{

**return** n;

}

**if**(!lookup.containsKey(n))

{

lookup.put(n, *fib3*(n-1,lookup) + *fib3*(n-2,lookup));

}

**return** lookup.get(n);

}

Reverse a linkedlist

public ListNode reverseList(ListNode head) {

if(head==null || head.next == null){

return head;

}

ListNode newHead = null;

ListNode current = head;

while (current != null)

{

ListNode next = current.next;

current.next = newHead;

newHead = current;

current = next;

}

return newHead;

}

Hashmap stuff

Capacity = number of buckets \* load factor

Load factor = number of entries / total size of map

Unix commands

Chmod

Mkdir

Cd

Kill pid

Rmdir

FIX Protocol is how finaical information is exchanged

Static typed lang is checked at compile time

A Btree is a generalization of a binary tree that aggregates data in blocks.