# Quiz Prep Week 11

## Concepts

Concepts covered in week 8 review (refer to that for practice)

- Big-O Notation
- Type Parameters
- Array List
- Linked List
- Inner Class

Concepts covered in week 9

- Hashing
- Maps

Concepts covered in Week 10

Exceptions

- Catching
- Throwing
- Working with Exceptions

## EMP Session Links

1. EMP 10-29
2. EMP 11-05
3. EMP 11-10

## Internet theoretical questions

1. Is the internet primarily wired or wireless?

▶ Spoiler!

Wired. Your laptop or phone may be wireless but they just connect to access points and the rest of the international internet is wired because it's far more energy efficient, reliable and cheaper.

2. What are different types of cables that make up the internet? How do we decide which ones to pick?

▶ Spoiler!

Many different cables exist. In class we looked at twisted pair CATV coaxial cables and glass based fibre optic cables. We decide based on the range we want (100m or kilometres), cost of cabling, bandwidth and speed.

3. Why do we have internet protocols? Who implements these protocols?

▶ Spoiler!

We need internet protocols because computers are bad at inferring what the incoming data (packets) is by context, so we define the structure of the packet and what to do with it. The protocols are just a set of rules (like interfaces) and it's up to the programmer to implement.

4. Let's say I have a phone which I use to open Wikipedia on my Firefox browser, which is the server and which is the client?

▶ Spoiler!

The phone running firefox is the client and Wikipedia has the server (in practice they have a giant cluster of them).

5. What is an I.P Address? What is your current IP Address? What information does it have?

▶ Spoiler!

IP Addresses are a unique identifier of computers within a network. The address can indicate a lot about your location and where you are connecting as well.

6. When do we use `GET` and when do we use `POST`?

▶ Spoiler!

We use `GET` when we want to recieve data (for example a news article or online profile) and `POST` when we want to send data (like creating a new news article or updating your online profile)

## Concept Overview & Practice

1. Assuming there is class Laptop that contains a constructor that takes in one String input which is the name of the company, then consider the segment of code below:

```java
Laptop a = new Laptop("Apple");
Laptop b = new Laptop("Apple");
Laptop c = new Laptop("Microsoft");

System.out.println(a.hashCode()); // Line 1
System.out.println(b.hashCode()); // Line 2
System.out.println(c.hashCode()); // Line 3
```

Which two lines will print the same output?

▶ Spoiler!

Neither. Since a,b and c are all different instances, they'll all have a unique hashcode!

2. What Java objects have access to the built-in hashcode function?

▶ Spoiler!

All of them!

3. Write a function named oddOrEven that takes in an array of integer inputs and maps them to a True or False value based on whether its even or odd respectively. This function should return this map.

▶ Spoiler!

```
public Map<Integer, Boolean> oddOrEven(int[] arr) {
   Map<Integer, Boolean> map = new HashMap<>();
   for (int i : arr) {
      if (i % 2 == 0) {
       map.put(i, true);
      } else {
         map.put(i, false);
      }
   return map;
 }
```

4. Write a function named fractions, which takes in two int arrays of the same length and returns and array of integers that has the values from the first array divided by second. (Keep in mind that when you divide a number by 0 java will throw an ArithmeticException, so you should take care of that!)

▶ Spoiler!

```
public int[] fractions(int[] a, int[] b) {
   assert a != null && a.length != 0 && b != null && b.length != 0;
   int[] output = new int[a.length]; //or b.length
   try {
      for (int i = 0; i < a.length; i++) {
         output[i] = a[i]/b[i];
      }
      return output;
   } catch (ArithmeticException e) {
       System.out.println("Cannot divide by zero!");
   }
   }
```

5. Write a function isAscending that takes in an int array, and returns true if the array is in ascending order, but

throws an IllegalArgumentException otherwise

▶ Spoiler!

```
public boolean isAscending(int[] arr) {
   assert arr != null && arr.length != 0;
   for (int i = 0; i < arr.length - 1; i++) {
     if (arr[i] > arr[i+1]){
      throw new IllegalArgumentException("Array is not ascending");
     }
   }
   return true;
 }
```