CYDEO

Java Interview Questions

How would you rate your Java skills out of 10?

Talk about your understanding of programming concepts

Being around 7 or 8 is good. There is also more to learn, especially for the topics that aren't needed in the work for an SDET.

Empathizes your foundation or core understanding

Being able to read and use any java library and even reading code from other languages

Comfortable to take on any task and adequate understanding to be able to google anything I'm searching for and get the idea of any code I see

Give examples of using java outside of the testing scope

Why do you use Java instead of any other language?

The tools/libraries we use for testing are in java

It was the first language I was introduced too

It is a high level language

The more I use it the more I've come to enjoy programing in java

Having java as my first language also make it easier to understand low level languages and other popular languages are also easy to understanding coming from a java background

From a technical perspective java is great because it is platform independent. It can be written on one machine and run on another

What version of java are you using and why that version?

Java 8 has been the LTS version used for many years, so many companies still use it

Java 8 is not getting updates anymore, so there is a shift towards java 11 which is also a popular LTS version

In the end depends on the team and project

The versions add new features and update, but none of the updates have affected my main work flow

Do you only have experience using java for testing or have you used it for other areas like development?

This is open ended, you can say no I have only used Java for testing
If you say no you may want to emphasize your passion for testing goes being the passion for programming. That being said there doesn't always need to be passion, but you should show that you are interested and engaged in the work

Or if you are interested in some other things you can give specific examples

For example:

discord bot, working with excel, Spring framework(development), Android app (development – Kotlin), Minecraft

How many years have you been using java for testing?

This is open ended and relates to your resume

You can mention I have been using java for x years

You could have been using java before you started testing

Or you could have been doing manual testing for x years and then started using java for automation

What is the difference between the Heap and the Stack?

Stack holds: method frames, local variables, references

Heap holds: objects, String pool

Java handles memory on its own

Can talk about garbage collection: used to removes objects that aren’t needed anymore

What is the Java String Pool?

The String pool is a special memory location for String literals. Sequence of characters are used often

It provides better performance and memory management

Doesn’t create duplicate values, allows references to the same objects

String created using the new keyword are created in the heap directly like other objects

What are the differences between a class and an object?

A class is a blueprint
An object is the physical result of the class
(blueprint → building)

An object is the instance of a class

A class defines instance variables and methods
An object of a class gets a copy of the instance members

Can you describe how you would reverse a String in java?

When answering any coding question we need to talk about the concepts we will use and the general approach we would take to solve the task

To reverse a String we could use StringBuilder's reverse in method

But if they want to see our own approach and test our problem solving skills as well as delivery of our thought process then we should take it out in steps

Use a loop that goes from last index to 0
Read each char
Concatenate to a String
When loop is done we have the reversed String

What are the lengths of an empty String and a null String?

An empty String has no characters, so the length would be 0

A null String means there is no String object.
null is a placeholder for objects

If we tried getting the length on the null we would get NullPointerException

How would you reverse a sentence without reversing each individual word?

If we are given a sentence I assume each word is separated by a space so I would split the sentence by the spaces to get each word into a String array

Now that I had each word separated I would just need to reverse the elements of the array

Starting from the last index of the array read the elements
Concatenate each element into the order we wanted
In the end we have the original sentence in reverse, but the words themselves would not be reversed

What is the difference between continue and break in loops?

The break statement will stop the loop as soon as it is run

The continue statement will stop current iteration and go to the next iteration. This means any code after the continue will be not executed

These statements shouldn’t be used too much

How can you access the last element in an array?

There can be questions like this which you have done many times but it is also important to be able to explain the concept in your own words

Indexes start from 0
And we know how to always check the number of elements: arr.length

We can always do length – 1 to read the last index no matter how many elements there is

unless the array is empty

So doing arr[arr.length -1] can read the last index

Similar idea is applied to other index based topics

How would you find the max value in an int array?

The immediate thought is to sort the array and read the last index because that will be the biggest value

But, if we can't use built in code and need to explain our own logic we can break it into steps

Create a variable to track the biggest number. The first element can be used as the initial value
Then iterate through all the other elements. During each iteration we would compare the element with the biggest number variable we declared in the beginning.
Whenever the current element being iterated was bigger than the previous biggest number we could reassign our biggest number to the that number
This checking would continue until the end of the loop and at that point the biggest number variable would be holding our max value

How can you find the second biggest number in an array?

The quick way is to sort the array and take the element at length – 2 index.

What if there is multiple of the max number? Then we would need to iterate though and find the number from the end that doesn't match the max number

If we needed to do this all ourselves without sorting we could keep 2 variables to hold the max and second max values.

We would iterate through the numbers in the array and check two conditions:

the first condition would check if we found a new max number. In this case the old max would become the second max value and the new max number would be changed to the current element

the second condition to check would be if the current element was bigger than the current second max but less than the max number and in that case we would update our second max variable with that current element

By the end of the loop we would have our max and second max numbers

Describe how you would sort an Array, without using built in methods

Explain each step you would take to sort

For example the bubble sort is a popular algorithm (for testing we don't study algorithms)

To implement the algorithm I would declare a loop that iterates through each index of the array
Then I would create a nested loop that will start from 0 and go until the end of the array, but with some restations based on the next steps
In the inner loop I would need to compare the 2 elements next to each other. If the current element was bigger than the element next to it (index + 1) and I would swap the position of the two elements
By doing this I will be able to move the bigger numbers to the end and keep doing this for each index of the array

Can you give some examples of utility methods you created and using them in your framework?

Find specific examples of methods from your framework. Preparing how you would describe the utility method will make it easier to answer these questions

Common methods:

getDriver() – util for a singleton driver used to interact with a browser, designed to work for multiple different browsers

createConnection() – util to connect to a DB

runQuery() – runs a sql query and get the results

hover() – moves the mouse to a defined web element

wait() – many wait methods so elements do not cause exceptions

page loading, element clickable, element visible

Any action that requires a couple steps and can be done multiple times can be made into a utility method

What is the difference between static and non static methods?

Static members are accessed by the class name
Instance members are accessed by an object of the class
(still need to pay attention to access modifiers and importing for both)

So the static methods can be accessed easily in the project by giving the class name and method name
Static methods are used in utility classes because the reusable function needs to be easy to use

Instance, or non static methods, need an object first to access them. These methods also have unique copies for each object of the class

Static methods have a single version

Can a class be static?

This is a topic we didn't go into much

Can a class be static – yes

To declare a static class it must be defined in another class first. It will be a nested class

A static nested class allows the inner class to be accessed with an instance of the outer class

outer.inner

In general these classes are used to improve readability, but I don't see a use for them in testing – opportunity to ask follow up: How does your team use static classes in your framework?

Tell me the differences between Static and Instance Variable/Block?

A static variable belongs to the class and is access from the class name. There is a single copy of this variable

The static block can access the static members and is executed once, before anything else, when the class is loaded(used) for the first time

An instance variable belongs to the objects of a class, so in order to access them an object must be created first. Each object gets a copy of all the variables

The instance block can access the instance members and is executed any time an object is created

Can a class be final?

Yes classes can be final

A final class cannot be inherited

It can be a sub class, but cannot be a super class

For example the String class is final

Tell me the difference between local variable and instance variable?

A local variable can only be accessed within the block of code it is defined
The scope of the local variables goes from the opening bracket to the closing bracket

An instance variable has a larger scope because it is not restricted to one code block. Instance variables are accessed through an instance of a class

When there is a similar name between the variables java priorities the local variable

Q: what about global variables?
java doesn’t have global variables

Can you explain what access modifiers are?

Access modifiers are keywords that define the accessibility of a variables, method, or class

Using public, protected, default, or private you can limit where that code can be accessed

public – whole project
protected – inherited to whole project
default – same package
private – same class

If you were overriding a protected method would you be able to change the access modifier to public?

Yes, it is possible to change the access modifier to be more accessible when overriding. public is more accessible than protected so it is allowed

The access modifier would also stay protected

It could not be changed to default or private because those are less accessible

Lets say you had a class A that has some private variables, and then you were working in class B, how would you access the variables from class A?

If the variables in class A are private they would not be accessible outside of the class

If we needed to access the information in those variables we would usually create getter and setters methods

A getter method is what we need in this case. A public method that can return the value of the variable we wanted access to

This is the idea of Encapsulation for controlling how our information is used by giving indirect access

What is the difference between protected and default?

Without inheritance these two access modifiers work the same way. They give accessibility within the same package

Default always gives access in the same package, and with inheritance it allows the information to be passed to sub classes in the same package

Protected on the other hand has a different function in inheritance. Even though it behaves like default normally, protected information can be inherited anywhere in the project. The sub classes can inherit when public or protected access modifiers are used

Explain the main method

The main method allows java code to be run

The JVM searches for a main method to start execution

The static keyword allows the method to be called without needing an object

The String array arguments also allow information to be passed in during execution from command line

Can we execute java programs without using main method?

No, the main method is the main starting point for java code

There is some small tricks that can be done, like using a static block to run code, or a command like jshell to run code, but those are not practical uses

When it comes to running java the main method is needed.

There is some libraries that create the ability to run code without having to write the main method ourselves, but internally the main method is still the start point for execution.

For example @Test annotations in Junit allows us to run the tests, but internally it is called a main method

Tell me the difference between a method and a constructor?

A method is a block of code that needs to be called somewhere

A constructor is a special method that is automatically invoked whenever an object of a class is created

A method can be called as many times as needed, but the constructor will run only once for each object.

Constructors also do not have a return type and the names needs to the same as the class name

A default constructor is always given

What is constructor chaining?

Constructor chaining is a constructor calling another constructor

this() is used to call the other constructors

Chaining can be helpful because it allows reusability of code in the overloaded constructors

It is clear which constructors needs to run based on the arguments given

A super class constructor can also be called using super()

What is recursion?

Recursion is a when a method calls itself

The code continues to run in a simple pattern until the base case

It can be useful in some problem solving tasks, but haven't seen it being useful in testing

- opportunities to ask if they use it in testing and how

Commons task: Fibonacci, factorial

What is OOP and how have you used them in your project?

Object orientated programming concepts help define rules and structures for how the code should be created

These ideas help to design how classes should be created to be reusable, readable, and maintainable

Sometimes we don’t need to apply these concepts because testing has a different focus, but the libraries we use follow these concepts

Encapsulation: Driver, Config Reader
Inheritance/Abstraction: Collections framework, Selenium, JDBC
Polymorphism: WebDriver, Collections, Methods

Why are OOP concepts important for programming?

These concepts create an agreed upon structure and rules to follow in order to maintain reusable code

Others can easily break down code they have not seen before because they follow set patterns

Without encapsulation there wouldn’t be a way to filter information or control which code should be accessible. Could lead to issues
Without inheritance there would be much more repeated code
Without abstraction it would be more difficult to set up common ideas and would be more difficult to manage
Without polymorphism object could not be flexible and the classes would need to be defined in more fixed and complex structures

What is Encapsulation and how have you used it?

Encapsulation allows the programmer to define how a class should be used

Encapsulation can be defined in a couple ways:

- defines which information can be accessed in directly
- grouping data and methods into a unit

Traditionally a class is encapsulated when the variables have private access modifier and public getter and setter methods

Partial encapsulation can also be achieved by only providing a getter or only providing a setter

Examples: Driver Util, ConfigReader Util,
@FindBy - uses method from By class to define an object to achieve POM
JSON - pojo classes give access to the data

What is the keyword: this, and how would you use it in a static context?

This is a keyword that represents the instance of a class

It can be used in the class are a reference to the object of that class

It is often used to differentiate between instance and local variables

It cannot be used with static because it belongs to the instance level not the class

How do we achieve multiple inheritance in java?

In java we cannot have multiple inheritance between classes
Meaning a class cannot have more than one parent

Interfaces are not classes, so they do not follow the same rules. We are able to inherit multiple interface to each other. We are also able to implement multiple interface to a class

This allows us more flexibility to create abstract ideas and implement them later

In the scope of inheritance how would the child class access any of the variables from the parent class?

If the access modifiers allow the variables will be inherits from the parent class to the child class
This those cases the variable can be access directly within the class or from an instance of the class

In some other cases we can use the keyword super to refer to the instance of the parent class

If there is variable hiding done, which is created a variable with the same name is the child class, this we can access the one from the parent with super.variable

super.method() can call a method implementation from the parent class

In a similar way super() is used to call the parent class constructor

What is data hiding?

This idea is tied with encapsulation

Data hiding is done, using access modifiers, to prevent direct access to a variable

This allows control of how the information can be used

Getters and setters could allow access indirectly but in a restricted way

Can you tell me the difference between IS A and HAS A relations?

IS A can define the inheritance relationship between two classes. When a sub class is inheriting a super call we describe the relationship as sub class IS A super class

HAS A is used to describe the relationship between classes in a different way. In this case they do not have an inheritance relationship but instead one class is given as a reference to the other class by being an instance member.

Example:

```
class Car{}

class Tesla extends Car {
        Battery battery;
}
class Battery{double watts}
```

Tesla is a Car
Tesla has a Battery

Tell me the difference between extends keyword and implements keyword?

extends is used to inherit from class to class
extends can also be used to inherit from an interface to an interface
interface can inherit many interface

implements is used to implement an interface to a class

extends is used for inheritance but implements is used for abstraction

What's the difference between this and super?

The this keyword is a reference to the current class
The super keyword is a reference to the parent class

this is mainly used to differentiate between instance variables and local variables
this() is used to call a different constructor in the same class

super gives access to variables from the parent class
And it can especially be useful when a method is overridden and you need to execute the original implementation from the parent class
super() is used to call the parent class constructor

What is the difference between method overloading & method overriding?

For any comparison question talk about the high level idea of each first. Don't start just listing rules and technical differences

Method overloading is done within the same class to create methods that share the same name, but have different parameters. There is already so many method we use and if we didn't have overloading we would need to remember and keep track of so many more.

think of indexOf, print methods

Method overriding is done between classes by allowing the implementation of a method, that was inherited, to be changed. Method overriding is needed for abstraction where abstract methods are defined and need to be implemented later

What are benefit of an Interface?

The biggest benefit of an interface is not being a class
Classes cannot have multiple parents, but interface do not follow those restrictions

This allows us to create abstract ideas and connect them together for a more reusable structure

Interface is used to define abstraction more than abstract classes because of this flexibility

Objects are able to use the interface as a reference which increases the flexibility of these interface even more

The libraries we use for testing and even within java itself use interface to define abstraction: collection framework, selenium

What are limitations of an Interface?

Interface are not a class

An interface cannot have a constructor

No blocks of code, like static or instance blocks are allowed

Instance members cannot be defined

Can only have static final variables

Tell me the difference between comparable and comparator interface?

We haven’t talked about this in class

These two interface are related to how objects should be sorted in data structures

The comparable interface allows you to define how the object should be sorted in relation to the other objects of the class
This is implemented in the class of the objects we want to define sorting order for
Comparable uses the compareTo() method

The comparator interface helps to define how objects of same or different classes can be sorted in relation to each other
This is implemented into a separate class, not in the classes of the object types
Comparator uses the compare() method as well as some others

How is abstraction achieved in java?

Abstraction can be achieved by creating an abstract class or an interface

An abstract class is a class that can define abstract methods, but cannot be instantiated

An interface is made in a java file, but instead of the class, the keyword interface is used
Interface have their own properties
An interface can also define abstract methods and cannot be instantiated

Interface is more flexible

What are the differences between final methods and abstract methods?

These methods are opposite ideas of each other

A final method cannot be overridden, so the original implementation can never be changed

An abstract method is a method with no implementation, or code body. It defines an idea without worry about how it is done. These methods are meant to be inherited and overridden to be implemented.

Another variation of the question: Can you make abstract final methods?

No these cannot the used on the same method

What is the difference between an abstract class and a non-abstract class?

An abstract class can create abstract methods, but it cannot be instantiated

Besides those two things an abstract class behaves the same as a normal class

A non-abstract class is just a normal class we are used to. It cannot hold abstract methods, so if any abstract method is inherited it must be implemented

The first non-abstract class in the inheritance of some classes is called the concrete class because it must implement the abstract methods

Describe the differences between an abstract class and an interface

An abstract class is still a class, so it is able to define instance members, constructors, blocks of code

A common tricky thought is that an abstract class cannot have a constructor, but it needs a constructor for the sub classes to be created. The super class constructor always need to be invoked for an object to be created

Abstract classes are also able to define non abstract methods if needed

An interface creates abstract methods by default and can only define constant variables
For an interface to define a method that has some code body, either a static method or default method needs to be created

How have you used abstraction in your framework?

In general you would not need to use abstraction yourself in the testing framework. It would only be useful if there was many team uses the same structure you set up

But even if we do not implement abstraction ourselves we are using it in the libraries we use

From the java library itself we use the Collections framework which is built on interfaces

In Selenium the WebDriver(interface), WebElement(interface), By class(abstract class)

In JDBC there are interfaces that allow connections to different databases: ResultSet, Connection, Statement

What is polymorphism?

Polymorphism allows objects to take different form through their references

An object is able to use a reference of its own class, any super class, or any interface that is implemented

Interface references are used often because of the structure inheritance and abstraction define

Methods and data structures become more flexible because of polymorphism

Important to note the reference must have access to use an instance member, but the execution happens on the object

Example: Selenium WebDriver

What is runtime polymorphism and static polymorphism?

These ideas talk about when the binding is done on the object.
Calling a method and executing the method are different and occur at different stages
A method can be called anywhere in the code, as long as the reference allows, but the execution happens during the runtime

These two concepts are talking about those different stages
Static polymorphism occurs during compiling. A reference must have access to variable or method otherwise the program will not compile. Method overloading is an example of static polymorphism

Dynamic polymorphism occurs during runtime. The value or implementation of a method is determine from the object. Method overriding is an example of dynamic polymorphism because the result is based on the implementation of the object, not the reference

How have you used polymorphism in your framework?

Polymorphism is used in almost all the code we use. Using the other references allows the code to be reusable and flexible

When using Collections the interface reference allows the data structure to follow the defined ideas

Methods can be created and called using different references. Some method are define with the Object class

List<Map<String,Object>> is one way we get data during testing and this allowed us to read the information from a DB and know use the data in java

What exception have you faced and how you did you handle them?

Think back to when you work on projects or were practicing and create a list of exceptions you faced. Try to be as unique as possible for different situations

Generic ones that occur: OutOfBoundsExceptions, NullPointerException, ElemenetNotFoundException, StaleElementException, ClassCastException

To resolve an exception two things can happen. Either something in your code was causing an exception so you fix your code to work properly, or it is not something in your control so you use a try catch to handle the exception and allow your program to continue

What is difference between runtime exception and compile time exception?

Throwable is the parent of Exception

Exception is the parent of all checked exceptions
Checked exceptions must be handled otherwise the program will not compile

RuntimeException is the parent of all unchecked exceptions
The RuntimeException class itself is a sub class of Exception
Unchecked exceptions occur during runtime, so they don’t need to be handled before running the program.

Both can be handled with a try catch

Give me an example of up casting and down casting?

References of an object can be changed with casting

Upcasting happens automatically and occurs when going from a sub class reference to the super class reference

Downcasting must be done manually when going from a super class reference to a sub class reference

An example of upcasting could be if we had the browser specific type and needed to run the test on a different browser as well we could need to cast up to the interface

An example of downcasting could be using the reference of a browser specific implementation to access unique methods

A common casting example is to cast the driver reference to the TakeScreenShot interface reference

When would you encounter a class cast exception?

Objects can have many types of references, and it is possible to cast between those references

But if you try to cast to a reference the object does not have an is a relation to the ClassCastException will be thrown

Meaning if there is no inheritance relationship between the object and the reference it will stop your program

You can only cast to and use references that have is a relation

Can a catch block handle multiple exceptions or only one at a time?

It is possible to do both

There can be as many catch blocks needed to handle the possible exception in the try block
When there is multiple catch blocks they should be in order of most specific type to least specific type

After java 7 it is also possible to use the or | operator to define multiple exceptions for the same catch block

Using the | operator can make the code readable, but if the code needed to run is different based on the exception they will need to be sperate catch blocks

When would you encounter a NullPointerException?

null is a placeholder for object. It is used when no object exists but the reference is created

NullPointerException occurs whenever trying to use an object, but no object is referenced

Ex:

```
String str = null;
str.charAt(0) → causes NullPointerException
```

Is the catch block mandatory?

A try that is created with a finally block does not need a catch block to compile, but not having a catch defeats the purpose of handling exceptions

A try without a catch would throw an exception if it occurs

A catch block is recommended to handle an exception so the program can continue execution instead of stopping

When does finally block not execute?

The finally block is meant to always be executed, regardless if there is an exception thrown or not

One way to force it to be skip is by using System.exit() which stops the whole program
This is not a valid scenario

Other times a finally block will not execute is situations that are beyond the control of the code itself

Finally was designed to always run and that’s how we want to use it

What is the difference between final, finally and finalize?

These are all different keywords that have their own functionality. No connection

final: variables cannot be changed, class cannot be inherited, and methods cannot be overridden

finally: an optional block used with try/catch statements. It is always executed at the end regardless if there is an exception or not

finalize(): a method called by the garbage collector to help clean the memory

→ Deprecated in java 9

What is the difference between the keywords throw and throws

These keywords are used in relation to exceptions

The throw keyword is used to cause an exception to occur in the program
The throw is used alongside an Exception object

```
throw new RuntimeException()
```

The throws keyword is used in the method signature to declare an exception may occur
throws does not handle an exception, but it can be used to compile the program in case a checked exception is possible
If that exception occurs it will still stop the program

throws is used by the developers of libraries. The exception is meant to be handled by the person using the code

What is garbage collection?

Java handles memory allocation and deallocation on its own

The heap memory is where objects are stored

So, the garbage collection is what cleans up the heap memory by removing object that are no longer needed

Objects that are eligible for garbage collection are objects that do not have a reference

Can you describe immutability and how it can be achieved

An object that is immutable cannot be changed after it is created
This means the value of an object is fixed after it is created, and to represent a different value a new object would need to be created

The final keyword can help create immutable objects by restricting variables from being changed and preventing a class being a super class

Using access modifiers and how an object's information can be accessed could also be a way to control the ability to change the object

The opposite of immutability would be mutability which would be object that can be changed after they are created

String is immutable

Tell me the differences between String, StringBuilder, and StringBuffer

These objects are used to represent character sequences and have slight differences

String objects can created in the String pool
StringBuilder and StringBuffer objects are created in the heap

String is immutable
StringBuilder & StringBuffer are mutable

StringBuilder is not thread safe
StringBuffer is thread safe

What is the Collection Framework in java?

The collections framework is built up of interfaces and classes that represent data structures with different algorithms to manipulate and handle data

The collections in the framework use different implementations to satisfy the algorithms being applied – So the programmer doesn't need to implement them

Only objects can be stored in these data structures

For testing we mainly use ArrayList & HashMap

Great example of Inheritance, Abstraction, & Polymorphism

Which collections have you used in your framework?

Go through and prepare a couple specific examples of when a collection was used.
Have a test case in mind and how you tested it

Common examples:
List for multiple WebElements
Set for window handles
Map for data from database or api

How would you find the unique elements from an array?

The first approach that comes to mind is converting the array into a List and using the removeIf method to remove all the elements that have a frequency of more than 1

A second approach that uses the Collections utility class. Iterate through each element of an array and use the frequency method to find only the elements that have a frequency of 1

Or if we needed to do everything ourselves we can iterate through each element of the array
Then run a nested loop that will compare each element to each other. We can count how many times the element matched with another element. At the end of each outer loop iteration we can check the counter and if it was 1 it is a unique element

How would you find if the ArrayList contained duplicate elements?

If we only care about having duplicates or not having duplicates then check the original size
Convert the List to a Set and all the duplicate elements will be removed
Compare the size of the collection before and after. If the same is the same there was no duplicate, but if the size is different some duplicates were removed

Another approach could be to sort the List and compare each element to the element next to it. If the element is ever the same as the element next to it there is duplicates

To do it all ourselves we would need to iterate through the List and use a nested loop to compare each element with all the other elements. By counting how many times they are the same we can determine if they are duplicate

When would you use an array or an ArrayList?

An array is useful when we have a fixed amount of data and do not need to add or remove data
If we don’t need to adjust the number of elements the array can hold that data with less memory

The array can also hold both primitive and non primitive types. ArrayList can only store object types

If we need to manipulate the data by adding or removing the ArrayList will be better
When we need more than just hold data the ArrayList can be helpful because it has methods

Both have useful utility classes Arrays and Collections

What are the differences between List, Queue and Set?

These are all interface in the Collection framework so they have designed idea around how to work with data

The List types are good for a flexible data structure that allows duplicates and has indexes to access specific elements that maintain some ordering

The Set types are good to store data that is unique without needing to worry how they are accessed. No duplicates or indexes

The Queue types follow a format of accessing elements in a FIFO approach. It maintains and uses data based on which ones were added first

What is the difference between LinkedList and ArrayList?

Both implement the List interface

The ArrayList uses arrays internally
The LinkedList uses nodes internally

By using or printing these two collections it is difficult to see the different. The differences come from the internal algorithms that are used in the implementations

ArrayList is better to store and get information
LinkedList is better to manipulate information. When there is a lot of changes to the position of the elements

What is the Map interface?

The Map interface does not inherit the Collection interface, but it is part of the java collection framework

This interface is for data structures that store information in a key-value pair format

The key-value pairs are called an Entry

The key is used to access the value
The key must be unique

What are HashMap and how do you use them in your framework?

HashMap implements the Map interface, so it stores data in a key/value format

Internally it uses a Hash Table algorithm

The order of the entries is not maintained

HashMaps are useful to store information from a database or api

In a data base there is columns that define different fields like name, address, city, etc, so those are stored into the map as the key and the value is the actual value of a person

Tell me the differences between HashMap and LinkedHashMap?

Both implement the Map interface so they are data structure that store information with key/value pairs

The main difference is how the entries are stored
HashMap entry order is not guaranteed
LinkedHashMap maintains the insertion order of the entries

The LinkedHashMap extends HashMap

What is the difference between a HashMap and HashTable?

The HashMap is the preferred data structure to store entry based data
The HashTable is a legacy class

The HashMap is not thread safe
The HashTable is tread safe

Both do not maintain the order of entries

HashTable cannot have a null key or value

What is thread safe?

Thread safety prevents an object from being accessed by multiple threads at the same time

Thread safety is achieved with synchronization

When an object is thread safe the threads would need to wait to access the object one at a time

Many legacy classes were thread safe because safety was more important in that time of development

How do you make your code thread safe?

Synchronization was used to make java code thread safe. The keyword synchronized is used in the methods

Using data that was immutable also achieved thread safety because there is never a concern of the data changing

There is also thread safe data structures that can be used (Vector, CopyOnWriteArrayList, ConcurrentHashMap)

What is multithreading and how would you use it as a SDET

Multithreading is executing multiple threads at the same time. It allows more resources to be used in order to save time

We can use it for parallel execution where we execute different test cases at the same time on different threads.

We could also run the same test cases, but on different environments or browsers at the same time

Selenium Grid allows multithreading

What iterator methods do you use?

The Iterable is implemented in the Collection objects

This defines the iterator method that returns the Iterator object that allows manually going through each element of the data structure

The for each loop works because of this iterator implementation

But if we needed to iterate through the data structure ourself we would use the next() and hasNext() methods

If we needed to remove an element we could use the remove() method

What is the difference between iterator and for/for each loops?

The iterator and for each loop work in a similar way
They both iterate through the elements one at a time
The for each loop doesn’t need any extra work. It goes through all the element of a data structure from beginning to end
The iterator needs use to the methods to iterate through one element at a time

The traditional loop can also be used to iterate through data structure, but only when we can read an element based on an index number

The iterator is useful to remove element without getting the ConcurrentModificationException

How do you iterate through a HashMap?

The Map interface does not extend the Iterable interface so Map objects cannot be put into a for each loop directly. The data itself is also different than other data structures

We need to use different approaches to go through the data

the keySet() method returns a Set of the keys. We can iterate through those keys, since they are a Set type, and access the value with the get() method

The values() method would allow us to go through all the values of the given map

The entrySet() is another method that allows us to iterate through each Entry. From each Entry we would be able to access each key and each value

What are Java generics?

Generics allow the types given to a class, object, method to be more flexible

The generics act like a parameter for the class
When an object is created from a class that has a generic type a type needs to be given in the diamond/angle brackets <>

Once an object is made when some defined type the methods or any other code in that class will know which type to use

Think of the Collections. When you define them we give the element type in the <> brackets. This is because the collections have a generic type and we are defining what type the elements will be for that collection

What are streams?

Streams allow us to work with large amount of data in a quick and easy way

Streams themselves are not data structures. They just allow us work with a lot of data

We can get a Steam from a data structure and then do some actions to the data

Some common steam methods: map, filter, count, forEach, reduce

This approach is quick, readable, efficient

How can you filter or map values?

Filtering and map are two functions that can be done in a stream

The filter method accept a Predicate argument and returns the data that match that Predicate

The map method takes a Function as an argument and returns the data after the given function is applied to them

stream().filter()
stream().map()

Predicate and Function are functional interface: interface that define one method. Those methods are implemented with a lambda expression