**Java Hackathon Theory**

**Q35. Write test cases for how to test just the withdrawing functionality from ATM ( Minimum 10 test cases required )**

1. We need to verify the ‘ATM Card Slot’ is as per the specification  
   2. We need to verify the ATM machine accepts card and PIN details  
   3. We need to verify the error message by inserting a card incorrectly  
   4. We need to verify the error message by inserting an invalid card (Expired Card)  
   5. Verify the user is allowed to do only one cash withdrawal transaction per PIN request  
   6. Verify the machine logs out of the user session immediately after successful withdrawal  
   7. Verify the message when there is no money in the ATM  
   8. Verify the cash withdrawal functionality by entering some valid amount  
   9. Verify the cash withdrawal functionality by entering an amount less than 100  
   10. Verify the cash withdrawal functionality by entering an amount greater than the total available balance in the account.  
   11. Verify the ATM machine successfully takes out the money.  
   12. Verify the ATM machine takes out the balance printout after the withdrawal

**Q36. Write to test scenarios to test Pencil**

1. Check is the shape and height of the pencil is as per the specification or not.
2. Check the darkness After writing something by a pencil.
3. Check the comfort while you write.
4. Check the thickness of the LED.
5. Check Are you able to write on paper, wall, and a few other places.
6. Check how often the LED is broken.
7. Check the color of the pencil.
8. Check after writing the text is readable or not.
9. Check the written text are erasable by a normal eraser or not.
10. Check that a normal sharpener can sharpen the pencil or not.

**Q37. What is JVM and explain me the Java memory allocation**

JVMis an engine that provides runtime environment to run the Java Code or applications. It converts Java bytecode into machines language. JVM is a part of Java Run Environment (JRE).

JVM is the one that actually calls the **main** method present in a java code.

Java applications are called Write Once Run Anywhere. This means a programmer can develop Java code on one system and can expect it to run on any other Java enabled system without any adjustment. This is all possible because of JVM.

The Java Virtual Machine divides the memory into Stack and Heap Memory.

The Stock Memory allocation in java is used for static memory and thread execution. The values contained in this memory are temporary.

Java Heap Space mainly used by java runtime, Java Heap Space comes into play every time an object is created and allocated in it.

**Q38. What is Polymorphism and encapsulation?**

Polymorphism ability of an object to take many forms. Most common use is in inheritance. Any function or method which is accepting parent class object ,it will accept child class object too. Any child object can play its role and of its parent role.

Encapsulation process of hiding implementation details from user, binding object state and behavior. For example if we want to access a private variable outside the class we use setters and getters.

**Q39. What is method overloading and Method over riding?**

Overloading ability of a class to have more than one method with same name and different method signature.

Either number of arguments or return type of arguments are different.

Return type of method will not form method overloading.

Overloading happens in same class.

We will come to know at compile time

Overriding comes into picture in inheritance.

Ability of child class have method with same name and same signature of parent class method.

Overriding happens in different classes that is in parent class and child class.

JVM can differentiate method only at runtime.

**Q40. Why string is Immutable?**

Cached String literals are shared between multiple clients hence its immutable

Since Strings are popular as the HashMap key, it's important for them to be immutable  
Used to open a network connection -host name port number.

Used as database URL for opening connection.

We can share between multiple threads since strings are immutable.

**Q41. What is the difference between String and String buffer?**

String is mutable, thread safe and hence slow

Can create in two ways with new keyword or without.Hence can be stored in pool or heap memory

Equals and Hashcode methods are overridden

String Buffer is mutable, thread safe hence faster than String

Can create only with new keyword hence always stored in heap

Equals and Hashcode methods are not overridden

**Q42. What is the difference between array and array list?**

Arrays are fixed in size, can hold only homogenous data types, no inbuilt methods, can have objects and primitive types, for loop only for iterating elements

Array Lists are growable, can contain homogenous and heterogenous data types, has many inbuilt methods, can contain only objects, can iterate using for loop, for each, different iterators

**Q43. What is the difference between hash map and Hash table?**

HashMap is non synchronized. It is not-thread safe and can't be shared between many threads, it allows one null key and multiple null values**,** it is fast, it is traversed by Iterator.

Hashtable is synchronized. It is thread-safe and can be shared with many threads. It doesn't allow any null key or value. It is a legacy class. It is slow. It is traversed by Enumerator and Iterator.

**Q44. What is a vector in Java?**

It is a legacy class. Main point is it grows by double size if we try to increase size later. It is synchronized. It is thread-safe and can be shared with many threads. It is slow. It is traversed by Enumerator and Iterator.

**Q45. What is set in java?**

Set is an interface present in the [java.util](https://www.geeksforgeeks.org/java-util-package-java/) package and extends the [Collection interface](https://www.geeksforgeeks.org/collections-in-java-2/), it is an unordered collection of objects in which duplicate values cannot be stored. It has many inbuilt methods inherited from the Collection interface and adds a feature which restricts the insertion of the duplicate elements.

**Q46. What is an abstract class?**

Is a class declared with abstract keyword

This class cannot be instatiated.

Abstract class can have both abstract methods and concrete methods.

We can have Abstract class without abstract methods.

A class with atleast one abstract method should be Abstarct

In order to use abstract class we need to extend class and implement all abstract methods if we don’t need to implement all abstract methods that child class also should be abstract.

**Q47. What is an interface?**

Is a type declared with interface keyword, must be public abstract (even though we specify)

Methods – will always be public abstract (even though we don’t specify)

Variables -will always be public static final (must initialize at time of declaration)

We can’t instantiate interface

Interface can extend multiple interfaces.

we can extend multiple interfaces

**Till java 1.8**

Interfaces will only have abstract methods

Class implementing interface should implement all methods os that interface

If a class implements two interfaces and both have same methods class need to implement that method only once

**Q48. Why Java is Platform independent?**

A programming language is called platform independent if that program can be run on any platform (operating system). Java is platform-independent because the same java program can run on any operating system.

If you write a code in Java, then the program will be sent to the compiler for compilation. The compiler creates a .class file that is readable for JVM(Java Virtual Machine). Each operating system will have different JVMs, these are responsible for Java to be a platform independent.

**Q49. What are access modifiers? Give me an example?**

Access modifiers in Java helps to restrict the scope of a class, constructor, variable, method, or data member. There are four types of access modifiers available in java:

Default – No keyword required, Private, Protected, Public

**Default**: When no access modifier is specified for a class, method, or data member – It is said to be having the **default** access modifier by default.

**Q50. What are java exceptions? Give me an example?**

An exception (or exceptional event) is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disrupted and the program/Application terminates abnormally.

Two times of exceptions are Checked exceptions and UnChecked exceptions

A checked exception is an exception that is checked (notified) by the compiler at compilation-time. (I/O exception)

A Unchecked exceptions are not checked at compile time but happens at runtime. (IndexOutOfBoundException)

**Q51. What is the difference between throws and throwable?**

Throws used in method signature, indicates that method may throws exceptions that listed after throws keyword.

Throwable is super class of all erros and exceptions in Java, is a member of java.lang package.

Java V M or throw keyword can throw exceptions that are either instances of throwable class or of its subclass.

**Q52. What is the difference between Error and exception?**

An Error is a subclass of Throwable that indicates serious problems that a reasonable application should not try to catch. Most such errors are abnormal conditions. The error indicates a problem that mainly occurs due to the lack of system resources.

Eg: OutOfMemoryException, System crash

An exception (or exceptional event) is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disrupted and the program/Application terminates abnormally.

Two times of exceptions are Checked exceptions and UnChecked exceptions

A checked exception is an exception that is checked (notified) by the compiler at compilation-time. (I/O exception)

A Unchecked exceptions are not checked at compile time but happens at runtime. (IndexOutOfBoundException)

**Q53. What is the difference between Error, throwable and exception?**

**An Error** is a subclass of Throwable that indicates serious problems that a reasonable application should not try to catch. Most such errors are abnormal conditions. The error indicates a problem that mainly occurs due to the lack of system resources.

Eg: OutOfMemoryException, System crash

**Throwable** is super class of all erros and exceptions in Java, is a member of java.lang package.

Java V M or throw keyword can throw exceptions that are either instances of throwable class or of its subclass.

**An exception** (or exceptional event) is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disrupted and the program/Application terminates abnormally.

Two times of exceptions are Checked exceptions and UnChecked exceptions

A checked exception is an exception that is checked (notified) by the compiler at compilation-time. (I/O exception)

A Unchecked exceptions are not checked at compile time but happens at runtime. (IndexOutOfBoundException)

**Q54. What are collection APIs, give me an example**

The Collection API is a set of classes and interfaces that support operation on collections of objects. These classes and interfaces are more flexible, more powerful, and more regular than the vectors, arrays, and hashtables if effectively replaces.

A collection is simply an object that groups multiple elements into a single unit.

we can call it as container also,

Example of classes: HashSet, HashMap, ArrayList, LinkedList, TreeSet and TreeMap. Example of interfaces: Collection, Set, List and Map.

**Q55. Null**

**Q56. Will java support multiple inheritance?**

When one class extends more than one classes then this is called **multiple inheritance**. Java doesn’t allow multiple inheritance to **avoid the ambiguity** caused by it. One of the examples of such problem is the **diamond problem** that occurs in multiple inheritance.

But Java support implementing multiple interfaces.

**Q57. What are the different types of interface? (Ans List, set, Queue)**

List interface is the child interface of Collection interface. It inhibits a list type data structure in which we can store the ordered collection of objects. It can have duplicate values.

List interface is implemented by the classes ArrayList, LinkedList, Vector, and Stack.

Set Interface in Java is present in java.util package. It extends the Collection interface. It represents the unordered set of elements which doesn't allow us to store the duplicate items. We can store at most one null value in Set. Set is implemented by HashSet, LinkedHashSet, and TreeSet.

Queue interface maintains the first-in-first-out order.Main methods are poll peek add remove

There are various classes like PriorityQueue, Deque, and ArrayDeque which implements the Queue interface.

**Q58. What are wrapper class? Give me an example?**

A Wrapper class is a class whose object wraps or contains primitive data types. When we create an object to a wrapper class, it contains a field and, in this field, we can store primitive data types. In other words, we can wrap a primitive value into a wrapper class object.

Int -Integer

Double-Double

**Q59. What is boxing and unboxing in Java? Explain with an example?**

Boxing – converting primitive data types to Wrapper class

Int i =9;

Integer J= new Integer();

J= Integer.valueOf(i);

Unboxing- converting Wrapper class to primitive data types

Int k;

K=J.intValue();

**Q60. Explain for each loop?**

For-each is another array traversing technique like for loop, while loop, do-while loop introduced in Java5.

It starts with the keyword for like a normal for-loop.

It’s commonly used to iterate over an array or a Collections class (eg, ArrayList)

for (type var: array)

{

statements using var;

}

**Q61. What are iterators, explain with an example?**

Iterators are used in [Collection framework](https://www.geeksforgeeks.org/collections-in-java-2/) in Java to retrieve elements one by one. There are three iterators.

Mainly there are 3 types of Iterators

Enumeration, Iterators, List Iterator

Iterators are most common because it is universal. 3 main methods hadNext(),next(),remove().

Iterator itrObj = LinkedListObject.iterator();

**Q63. What is multithreading, serialization and Generics in Java**

Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.  
  
Threads can be created by using two mechanisms:  
1. Extending the Thread class  
2. Implementing the Runnable Interface