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

**#1)** Verify if the card reader is working correctly. A screen should ask you to insert the pin after inserting the valid card.

**#2)** Verify if the cash dispenser is working as expected.

**#3)** Verify if the receipt printer is working correctly. Which means it can print the data on the paper and the paper comes out properly.

**#4)** Verify if the Screen buttons are working correctly. **For touch screen:** Verify if it is operational and working as per the expectations.

**#5)** Verify if the text on the screen button is visible clearly.

**#6)** Verify the font of the text on the screen buttons.

**#7)** Verify each number button on the Keypad.

**#8)** Verify the functionality of the Cancel button on the Keypad.

**#9)** Verify the text color of the keypad buttons. The numbers should be visible clearly.

**#10)** Verify the text color and font of the data on the screen. The user should be able to read it clearly.

Q36. Write to test scenarios to test Pencil

1. Verify that the text written with the pencil is readable/legible.
2. Verify that the user can write smoothly on different types/quality of paper surfaces.
3. Check that the darkness/color of the text written by pencil is as per the specifications.
4. Check the strength of the lead, it should not break when a specified(normal human) pressure is applied.
5. Verify that the text written by pencil can be erased by normal erasers.
6. Verify that the quality and strength of the pencil’s wood.
7. Check whether the outer body of the pencil is circular or some polygon shape.
8. Verify that the length and radius of the pencil are as per the specification.
9. Verify that the weight of the application is as per the specification.
10. Verify that the pencil can be sharpened easily by a normal sharpener.

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

JVM is Java Virtual Machine.

Heap is a section of memory which contains Objects and may also contain reference variables. Instance variables are created in the heap

If you create String object with New keyword it will create new object in Heap memory. String is immutable means references are changing not values.

Classes are like blue print And Objects are the real once.If there is no static declared, memory will be created as many times objects are called.Static means they share the single memory for all the instances.

Q38. What is Polymorphism and encapsulation?

**Polymorphism** means "many forms", and it occurs when we have many classes that are related to each other by inheritance. There are two types Static which will happen compile time. Method overloading is Static Polymorphism. Dynamic polymorphism will happen at run-time. Method overriding is Dynamic polymorphism.

Encapsulation is binding of object state and behavior. We are hiding the implementation of methods from user. In **encapsulation**, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class.

Q39. What is method overloading and Method over riding?

Method overloading occurs when two or more methods in one class have the same method name but different parameters. Overloading is done in same class. Overriding means methods having same name and parameters. One method is in parent and other is in child class.

Q40. Why string is Immutable?

String is a special class. When you modify, new string is created. References are changing not values. For example

String s1 = **"Hello World"**;

String s2 = **"Hello World"**; if(s1 == s2) is True() ;

two different variables are pointing to same String object from the pool, thus saving crucial memory resource.

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

String is immutable. When you modify, new string is created. Its thread safe. Deadlock will not arise

Slow in process.

String Buffer is mutable. If you modify any string, existing string will get modify.

Thread safe. Deadlock will not arise. Slow but faster than string.

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

Arrays are static in length. Its very difficult to add or remove items. Not many inbuilt functions.

Arraylist is dynamic. Elements are inserted or accessed by their position in the list. You cannot use

Primitive data type in arraylist.

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 without proper synchronization code whereas Hashtable is synchronized. It is thread-safe and can be shared with many threads.  
HashMap allows one null key and multiple null values whereas Hashtable doesn’t allow any null key or value.

Q44. What is a vector in Java?

The **Vector** class implements a growable array of objects. **Vector** implements List Interface. Like ArrayList it also maintains insertion order but it is rarely **used** in non-thread environment as it is synchronized and due to which it gives poor performance in searching, adding, delete and update of its elements.

Q45. What is set in java?

Set is a collection that can have duplicate values.

Hashset – one null value+no ordering of items

LinkedHashset – one null value+Insertion order is maintain

Treeset – No null value +ascending order

Q46. What is an abstract class?

Abstraction is a process where you show only relevant data .Abstract class can have abstract methods (method without body) and concrete methods.Abstract class cannot be instantiated.

When you have abstract method inside class ,declare class abstract.Method needs to be overridden.

When you don’t want to override abstract method, declare class abstract.

Q47. What is an interface?

Interface provides full abstraction. To access the interface methods, the interface must be "implemented" by another class with the implements keyword (instead of extends). The body of the interface method is provided by the "implement" class. Interfaces **cannot** be used to create objects. Interface methods do not have a body - the body is provided by the "implement" class

On implementation of an interface, you must override all of its methods. Interface methods are by default abstract and public. Interface attributes are by default public, static and final. An interface cannot contain a constructor (as it cannot be used to create objects).

Q48. Why Java is Platform independent?

When the Java program runs in a particular machine it is sent to java compiler, which converts this code into intermediate code called bytecode. This bytecode is sent to Java virtual machine (JVM) which resides in the RAM of any operating system. JVM recognizes the platform it is on and converts the bytecodes into native machine code. Hence java is called platform independent language.

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

As the name suggests 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:

1. Default – No keyword required
2. Private
3. Protected
4. Public

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

A Runtime error is called an **Exceptions** error. It is any event that interrupts the normal flow of program execution. **Example** for **exceptions** are arithmetic **exception**, Nullpointer **exception**, Divide by zero **exception**, etc. It belongs to throwable class.

Q52. What is the difference between Error and exception?

Errors you cannot handle. Example memory crash, out of memory.

Exceptions can be handled during runtime and continue the execution process.

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

Errors you cannot handle. Example memory crash, out of memory.

Exceptions can be handled during runtime and continue the execution process. We use try and catch block.

Throwable is a class and error and exceptions are parts of throwable class. Use throws keyord in main method to catch checked exceptions Ex: IOException.

 Q.54 What are collection APIs, give me an example

Collectionis a framework.Collection of interfaces and classeswhich helps in storing and processing the data efficiently.We can perform all operations on data such sorting,insertion,manipulation.

Q56. Will java supports multiple inheritance. You extend only one class but can implement many interfaces.

Q57. What are the different types of interface?

List, set, Queue

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

A **Wrapper class** is a **class** whose object wraps or contains primitive data types.

Example: Integer,Character, Boolean,Double

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

Boxing is conversion of primitive data type to the object of their corresponding wrapper class.

Integer number = 10;// autoboxing create object of Wrapper class.It converted into primitive type at run time.

Int i = number; //unboxing

Int i ; Number = i //unboxing

Q60. Explain for each loop? Use instead for for loop to iterate through array values.its use to set and print array values.

For( int i : Arrayname)

{

Syso(i);

}

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. Enumeration,Iterator,ListIterator.

Iterator<Integer> it = ll.iterator();

While(it.hasNext())

Syso(it.next());

Q62. write an algorithm to reverse first 3 numbers, then next 3 numbers, then next 3 numbers, the number will be based on var k. Array = [3,2,4,7,0,3,1,5,8, 4]    k=3    OutPut = [4,2,3,3,0,7,8,5,1,4]

Write outer loop for number of passes int k =3.Three passes are reuired.Last will remain the same.

Pass 1 : arr =[3,2,4,7,0,3,1,5,8, 4]

Left=0,Right= 2

While(true)

Swap the array from[0,2]

o/p = arr =[4,2,3,7,0,3,1,5,8, 4]

pass 2: Left=3,Right= 5

While(true)

Swap the array from[3,5]

o/p = arr =[4,2,3,3,0,7,1,5,8, 4]

pass 2: Left=5,Right= 7

While(true)

Swap the array from[5,7]

o/p = arr =[4,2,3,3,0,7,8,5,1, 4]

final o/p : arr =[4,2,3,3,0,7,8,5,1, 4]

Q63. What is multithreading, serialization and Generics in Java

**MULTITHREADING in Java** is a process of executing two or more threads simultaneously to maximum utilization of CPU.

**Serialization** is a process of converting an Object into stream of bytes so that it can be transferred over a network or stored in a persistent storage.

**Generics** means **parameterized types.** An entity such as class, interface, or method that operates on a parameterized type is called generic entity. User defined generic classes are created.