What is a Class?

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. A class in java can contain Field,Methods and Constructors,Block of code.

How to declare a class:

```
public  class FirstProgram{

                  --------------

               --Block of code----

                      }
```

here,

public ---> AccessModifiers

class ---> Keyword

FirstProgram ---> ClassName

2.What is an Object?

1. An Instance of a class is called Object.
2. An object is a real-world entity.

Syntax for Object:

ClassName objectName=new ClassName();

3.Explain about the main method in java

Main() method is the entry point of any standalone java application. The syntax of main method is **public static void main(String args[]){---}**. Main method is public

and static so that java can access it without initializing the class. The input parameter is an array of String through which we can pass runtime arguments to the java program.

public : Public is an access modifier, which is used to specify who can access this method. Public means that this Method will be accessible by any Class.

static : It is a keyword in java which identifies it is class based. i.e it can be accessed without creating the instance of a Class.

void : It is the return type of the method. Void defines the method which will not return any value.

main : It is the name of the method which is searched by JVM as a starting point for an application with a particular signature only. It is the method where the main execution occur.

String args[] : It is the parameter passed to the main method.

4.Data Types in Java

Data types specify the different sizes and values that can be stored in the variable. There are two types of data types in Java.

1. **Primitive data types:** The primitive data types include boolean, char, byte, short, int, long, float and double.
2. **Non-primitive** data types: The non-primitive data types include **Classes, Interfaces,** and **Arrays**.

5.What are the access modifiers available in java

The access modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying the access modifier on it. There are four types of Java access modifiers:

Private: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.

Default: The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.

Protected: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.

Public: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

A java class can only have public or default access modifier.

Access Modifier	within class	within package	outside package by subclass only	outside package
Private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y

6.Variables:

A variable is a container which holds the value while the Java program is executed. A variable is assigned with a data type.

There are three types of variables in java

1.Local Variables

2.Global Variables

3.Static Variables

7.Difference between Local Variable and Global Variable?

Local Variable	**Global Variable**
If you are declaring the variables inside a method, That is called as Local Variable	If you are declaring the variables inside a class, and outside of the method That is called as Global variable.
Should be Initialized	May or may not be Initialized
If you are declaring the local variable, it will be accessing only with in a method, but it can not accessed outside the method.	If you are declaring the Global variable, it will be accessing any where of this class.
For local variable access modifier can't be used.	For Global variable access modifier can be used.

8.Can we overload main method in Java

Yes, we can have multiple methods with name “main” in a single class. However if we run the class, java runtime environment will look for main method with syntax as public static void main(String args[]){---}.

9.Why is Java a platform independent language?

Java language was developed in such a way that it does not depend on any hardware or software due to the fact that the compiler compiles the code and then converts it to platform-independent byte code which can be run on multiple systems.

The only condition to run that byte code is for the machine to have a runtime environment (JRE) installed in it.

10. What is the difference between Break and Continue Statements in Java?

Both “break” and “continue” are the ‘jump’ statement, that transfer control of the program to another part of the program. Java supports three jump statements ‘break’ ‘continue’ and ‘return’. The main difference between break and continue is that break is used for immediate termination of loop whereas, continue terminate current iteration and resumes the control to the next iteration of the loop.

Break:

1.It terminates the execution of remaining iteration of the loop

2.'break' resumes the control of the program to the end of loop enclosing that 'break'.

3.It causes early termination of loop.

4.'break' stops the continuation of loop.

5.'break' can be used with 'switch', 'label'.

Continue:

1.It terminates only the current iteration of the loop.

2.'continue' resumes the control of the program to the next iteration of that loop enclosing 'continue'

3.It causes early execution of the next iteration.

4.'continue' do not stops the continuation of loop, it only stops the current iteration.

5.'continue' can not be executed with 'switch' and 'labels'.

11.**What is an Array? How will you declare an array in java?**

a) Array is a collection of similar data types.

b) It can not have different data type. It can hold both primitive types (int, float, double) and object references.

c) It is fixed in length i.e static in nature.

d) Arrays are created on the heap memory not on the stack.

e) Accessing an invalid index of an Array will cause exception.

You can declare an Array in java by the following way :

```
dataType[]    arrayVariableName  = new dataType[arraySize];
```

for example for int data type, you can declare an int array as :

```
int[]  temp = new int[256]
```

Array literal:

```
Int[ ] num={1,2,3,4,5};
```

12.**When will you get ArrayIndexOutOfBoundsException?**

ArrayOutOfBoundsException is thrown when an attempt is made to access the Array with illegal index. For example, illegal index means if the index is either negative or greater than or equal to the size of the Array.

For example,

```
public class ExceptionExample{
    public static void main(String args[]) {
           int[]  rollNumber = { 23, 17, 20, 29, 30 };
           /* Index below is greater than the size
           of the given Array */
           int element = rollNumber[6];
           System.out.println(element);
    }
}
```

What is String in java? Is it a Data Type?

String is a Java Class (part of JDK) and it is not data type like int, boolean, char etc. It is a sequence of characters and enclosed with the double quotes (" ").

For example:

String txt = "TestLeaf";

What are the different ways to create the String Object in java?

There are two ways to create the string object, by string literal and by new keyword.

i) String as literal like:

String s = "TestLeaf";

ii) String as Object like:

String word =new String("TestLeaf");

What is the Difference between .Equals and ==?

Both equals() and “==” operator in Java are used to compare objects to check equality. But there are certain differences between them.

a) .equals() is a method and == is an operator.

b) == operator is used for reference comparison (address comparison) and .equals() method for content comparison i.e., == checks if both objects point to the same memory location whereas .equals() evaluates to the comparison of values in the objects.

c) If a class does not override the equals method, then by default it uses equals(Object o) method of the closest parent class that has overridden this method.

Eg:

```
public class Test {
 public static void main(String[] args)
{
String s1 = new String("TEST LEAF");
      String s2 = new String("TEST LEAF");
System.out.println(s1 == s2);
      System.out.println(s1.equals(s2));
   }
}
```

Output:

```
false
true
```

Explain Oops Concept?

Oops Standard for Object Oriented Programmimg Language, Java is One of the Object oriented Programming Language, We have four concept in oops

1.Abstraction

2.Inheritance

3.Encapsulation

4.Polymorphism

Abstraction

Abstraction is the process of hiding certain details and showing only essential information to the user.
Abstraction can be achieved with either **abstract classes** or **interfaces**

The **abstract** keyword is a non-access modifier, used for classes and methods:

- **Abstract class:** is a restricted class that cannot be used to create objects.
- **Abstract method:** can only be used in an abstract class, and it does not have a body. The body is provided by the subclass (inherited from).

Example Selenium: By

Example Java: Abstract Class

Interface:

Interface is design, It have only UnImplement methods, Interface does not have body .

we can able to use public and default access modifiers but If you are creating a method with default access modifiers it will be considering as public access modifier.

We can not create object for Interface due to constructor should not Initialized.

Example Selenium: WebDriver, SearchContext, WebElement

Example Java: List, Set

Inheritance:

Inheriting variables and methods from super class to sub class

Types of Inheritance:

1,Single Inheritance

2,Multiple Inheritance

3.Multilevel Inheritance

Single Inheritance:

Inheriting Super class to sub class

Example selenium:

ChromeDriver extends ChromiumDriver

Example Java:

NoSuchElementException extends RuntimeException

MultiLevel Inheritance:

Example selenium:

ChromeDriver extends ChromiumDriver

ChromiumDriver extends RemoteWebDriver

Example Java:

NoSuchElementException extends RuntimeException

RuntimeException extends Exception

Polymorphism:

Poly → many

Morphism → multiple form of Implementation

Method Overloading / CompileTimePolymorphism / Static Binding

Same class same method but aruguments should be different,

Example Java: println();

Example Selenium:

driver.switchTo().frame(int Index);

driver.switchTo().frame(String NameOrId);

driver.switchTo().frame(WebElement element);

Method Overriding / RunTimePolymorphism / DynamicBinding

Different class with same method, but the class should be inherited

Example Selenium:

driver.quit()

Example java:

.equals()

Is multiple inheritance possible in java?

Java doesn’t allow multiple inheritance to avoid the ambiguity caused by it. One of the example of such problem is the diamond problem that occurs in multiple inheritance.

But possible in Interface level

Example: RemoteWebDriver implements WebDriver, JavaScriptExecutor, TakesScreenshot

What is an Exception? Why we are handle it?

Exception is an abnormal condition which will interrupt the normal flow of execution.

Why we are handle it:

To handle that exception, Enable the normal flow of execution

What is try, catch and Finally?

Try	The "try" keyword is used to specify a block where we should place exception code. The try block must be followed by either catch or finally. It means, we can't use try block alone.
Catch	The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.
Finally	The "finally" block is used to execute the important code of the program. It is executed whether an exception is handled or not.

Difference between Final, Finally and Finalize?

Final:

Final is a keyword and Non-access modifiers, we can able to use it in class, method and variable

class--> If you are declaring the final keyword in class level, we can not inherit with another class.

method --> If you are declaring final keywords in method level, The method can be overloaded but it can not overridden.

variables --> If you are declaring final keyword in variable level that variable can not be modified.which means we can not be overloaded and overridden

Finally:

finally is a block of code, Even though the exception occurs, The finally block will execute. we can not use finally block alone, It should be followed by try or catch blockFinalize:

finalize:

finalize() method in Java is a method of the Object class that is used to perform cleanup activity before destroying any object. It is called by Garbage collector before destroying the objects from memory. finalize() method is called by default for every object before its deletion.

Difference between throw and throws?

throw and throws both are java keywords

Throw	throws
we can able to throw the exception explicitly	throws it is a method signature which is used to handle that exception
At that time we can throw single exception only	At that time we can handle more than one exception

What is Constructor?

A constructor in Java is a **special method** that is used to initialize objects

The constructor name must be same as the class name.

A constructor must not have a return type.

The constructor is invoked implicitly.

The constructor name must be same as the class name.

Difference between Constructor and Method?

Constructor	Method
A constructor is used to initialize the state of an object.	A method is used to expose the behavior of an object.
A constructor must not have a return type.	A method must have a return type.
The constructor is invoked implicitly.	The method is invoked explicitly.
The Java compiler provides a default constructor if you don't have any constructor in a class	The method is not provided by the compiler in any case.
The constructor name must be same as the class name.	The method name may or may not be same as class name.

What is Constructor chaning?

Constructor chaining is the process of calling one constructor from another constructor with respect to current object.

One of the main use of constructor chaining is to avoid duplicate codes while having multiple constructor (by means of constructor overloading) and make code more readable.

What is static keyword in java?

The `static` keyword is a non-access modifier used for methods and variables. Static methods/variables can be accessed without creating an object of a class.

Difference between this and super keyword in java?

this	super
The current instance of the class is represented by this keyword.	The current instance of the parent class is represented by the super keyword.
In order to call the default constructor of the current class, we can use this keyword.	In order to call the default constructor of the parent class, we can use the super keyword.
It can be referred to from a static context. It means it can be invoked from the static context.	It can't be referred to from a static context. It means it cannot be invoked from a static context.
We can use it to access only the current class data members and member functions.	We can use it to access the data members and member functions of the parent class.