

1. What is Abstraction

Abstraction is process of hiding real implementation details from the user and just provides control access to the user with the necessary information.

2. Difference between abstract class and interface

Abstract class	Interface
1. Abstract class doesn't support multiple inheritance.	1.Interface supports multiple inheritance.
2. An abstract class can extend another Java class and implement multiple Java interfaces.	2. An interface can extend another Java interface only.
3. Abstract class can have final, static and non-static variables.	3. Interface has only static and final variables.
4. The abstract keyword is used to declare abstract class.	4. The interface keyword is used to declare interface.
5. Abstract class can provide the implementation of interface.	5. Interface can't provide the implementation of abstract class.
6. abstract class achieves partial abstraction (0 to 100%)	6.interface achieves fully abstraction (100%).

3.Advantages Of Abstract Class.

- a. Abstraction in Java avoids code duplication
- b. Abstract classes enable code reusability
- c.Changes to internal code implementation are done without affecting classes

4.Disadvantages of abstract class.

- a.we can not achieve 100% abstraction but we achieve 0-100% .
- b.Multiple inheritance is not possible.
- c.We can create a constructor of abstract class but we can not create object.
- d.We could not make abstract class as final or private.

5.can we make a class as a super and sub at a time?

Yes , we can make a class as a super and sub class at a time.

```
Class A{
```

```
}
```

```
Class B extends A{
```

```
}
```

```
Class C extends B{
```

```
}
```

6.can we make abstract class as a final or private ?

We can not make abstract class as final or private.

7.What is abstraction ? Where did you use abstraction in your framework?

Abstraction is process of hiding real implementation details from the user and just provides control access to the user with the necessary information.

Use::

1.WebDriver is an interface with the help of its reference we can use its abstract methods but we don't know the how it is implemented.

2. In Page Object Model design pattern, we write locators (such as id, name, xpath etc.,) and the methods in a Page Class. We utilize these locators in tests but we can't see the implementation of the methods. Literally we hide the implementations of the locators from the tests.

3.TakeScreenShot is an interface with the help of its reference we can use its abstract methods but we don't know the how it it implemented.

4.JavaScriptExecutor is an interface with the help of its reference we can use its abstract methods but we don't know the how it it implemented.

5.SearchContext is an interface with the help of its reference we can use its abstract methods but we don't know the how it it implemented.

8.What is Inheritance? Where you use Inheritance in framework?

“Inheritance is the process by which one class acquires the properties (instance variables) and functionalities of another class is known as Inheritance.”

--we use hierarchical inheritance.

Public class A

{

Public void test()

{

}

}

Public class B extends A

{

Public void demo()

{

}

}

Use::

We create a Base Class in the Automation Framework to initialize WebDriver interface, WebDriver waits, Property files, Excels, etc., in the Base Class.

We extend the Base Class in other classes such as Tests and Utility Class.

Here we extend one class (Base Class like WebDriver Interface) into other class (like Tests, Utility Class) is known as Inheritance.

9.What is Encapsulation?Where did you use Encapsulation in your framework?

“the process of binding States and behavior of an object into a single class is called Encapsulation”

Where ?

All the POM classes in a framework are an example of Encapsulation. In POM classes, we declare the data members using @FindBy and initialization of data members will be done using Constructor to utilize those in methods.

Why we follow encapsulation in POM class?

Because

- 1.maintainance and modification of all the elements with respective of POM class is easy.
- 2.in order to avoid staleElementReferanceException
- 3.We can restrict direct access of each and every WebElement.
- 4.We can achieve code optimization
- 5.dubbuging is easy.

10.What is polymorphism? where you use polymorphism in your framework?

“The process of same object is showing different states and different behaviours”.

It's the ability of the object to exhibit many forms.

In java we have two types polymorphism

1.compiletime polymorphism

2.run time polymorphism

Compile time polymorphism:

Its the compiler decides which implementation to be provided for method call statement during the compilation time is called as compile time polymorphism.

Types:

1.method overloading

2.constructor overloading

3.shadowing

Method OverLoading:

Declaring more than one method with the same names but different arguments with in the same class .

Ex:

Public class A

{

Public void add()

{

}

```
Public void add(inta,int b)
```

```
{  
}
```

```
Public void add(inta,String s)
```

```
{  
}  
}
```

Rules:

- 1.length/size of the formal arguments must be different.
- 2.data types must be different.
- 3.order of declaration.

Real-time Examples:

Facebook login we can login using mobile number and email both perform the same action landing to the homepage.

CONSTRUCTOR OVER LOADING:

Declaring more than one constructor within the same class but differ in the formal arguments.

```
Public class Book
```

```
{
```

```
Int price;
```

String name;

Private String author;

Book()

{

}

Book(int price)

{

this.price=price;

}

Book(int price,String name)

{

this.price=price;

 this.name=name;

}

Book(int price,String name,String author)

{

this.price=price;

 this.name=name;

this.author=author;

}

}

Runtime polymorphism:

Its process of providing the implementation to method call statement during the execution time is called as runtime polymorphism.

Types:

Method Overriding:

The process of providing new implementation to the super class method from the subclass.

Rules:

- 1.names of the methods should be same.
- 2.method signature and return type should be same
- 3.sub class should have same or higher visibility access modifiers than super class.
- 4.To achieve method overriding super class method should not be final.

Ex.

Public class A

{

Public void demo()

{

}

}

Public class B extends A

{

Public void demo()

{


```

}

}

Public class Driver

{

Public static void main(string[]args)

{

A obj=new A();

B obj1=new B();

Obj=obj1;

Obj.demo();

}

}

```

Where did you achieve method overriding in your framework?

In my framework I achieved runtime polymorphism in Webdriver Interface ,RemoteWebdriver and driver class.

Where did you achieve method overriding in realtime?

Any application updates(phonepe,whatsapp,...).