



### This calling

1. It is use in the case of constructor overloading
2. It should be written explicitly
3. It is use to call from one constructor to another constructor within class
4. It should be the first line in any constructor

### Super Calling

1. It is use in the case of inheritance
2. It can be written both ways
3. It is use to call from one one constructor of one class to another constructor of another class
4. It should be the first line in any constructor and we can't have combination of this and super calling statement in any constructor



## This Keyword Vs Super Keyword

### **This keyword**

1. Whenever global variable and your method parameter are same then we can use this keyword
2. This keyword can be written only in the case of non static method

### **Super keyword**

1. It is use only in the case of method overriding if you need parent class implementation along with child class implementation
2. Super keyword can be written only in the case of non static method

## Abstract Class

1. Abstract class can have abstract and non-abstract methods.
2. Abstract class doesn't support multiple inheritance.
3. Abstract class can have final, non-final, static and non-static variables.
4. Abstract class can provide the implementation of interface.
5. The abstract keyword is used to declare abstract class.
6. An abstract class can extend another Java class and implement multiple Java interfaces.
7. An abstract class can be extended using keyword "extends".
8. A Java abstract class can have class members like private, protected, etc.
9. It can have constructors and static methods also.

## Interface

1. Interface can have only abstract methods. Since Java 8, it can have default and static methods also.
2. Interface supports multiple inheritance.
3. Interface has only static and final variables.
4. Interface can't provide the implementation of abstract class.
5. The interface keyword is used to declare interface.
6. An interface can extend another Java interface only.
7. An interface can be implemented using keyword "implements".
8. Members of a Java interface are public by default.



### Throw keyword

1. Used within a method (or constructor)
2. Used to throw an exception explicitly
3. Can only throw a single exception
4. Followed by a throwable instance
5. Cannot be used to propagate checked exceptions by itself

### Throws keyword

1. Used with method (or constructor) signature
2. Used to declare exceptions
3. Can declare multiple exceptions
4. Followed by an exception class name
5. Can be used to propagate checked exceptions by itself



### Local Variable

1. Definition: It is present within any method
2. Scope: Within method
3. cannot classify them into static and non static
4. it will not have any default value
5. It has to be initialized before utilising them.

### Global Variable

1. It is present outside any methods but within the class.
2. From the beginning of class and til the end of clas
3. It can be static and non static
4. Global Variable will have its own default value
5. Can be utilised without even initializing it

## Final vs Finally Keyword

### Final Keyword

1. It can be used with variable, Class and methods.
2. Any variable which is final will not change its value
3. Any method which is final cannot be overridden
4. Any class which is final can never be inherited.

### Finally Keyword

1. It is keyword to be used in exceptional handling concept of Java.
2. Finally keyword to be used only after try and catch block.
3. Since we are not sure if try will execute or catch will execute but we are sure that finally will execute for sure.

## Arrays

1. Can store only homogeneous value
2. Arrays are known for its fixed size-we cannot increase or decrease the size of an array at run time
3. Arrays is a in built feature of java but we have to develop the logic

### Syntax:

```
String name[]=new String[3];
```

or

```
String name[]={"RAM","Vishnu","Sham"};
```

## Collection

1. It can store both homogeneous and heterogeneous value.
2. here we can increase or decrease the size of a collection at run time
3. Collection framework is an API which provides classes,methods and interfaces in it

### Syntax:

```
ArrayList a1=new new ArrayList();
```

## FindElement Vs FindElements

### FindElement

1. Return type is WebElement
2. If element is not found it will give NoSuchElementException

### FindElements

1. Return Type is List<Webelement>
2. If element is not found it will give empty list



## Implicit Wait

1. Here we don't specify any condition
2. If specify time is over then it will throw NoSuchElementException
3. Once written it can be use for all FE and FE's in the script

Syntax:

```
driver.manage().timeouts.implicitly  
wait(10,TimeUnit.Seconds)
```

## Explicit wait

1. Here we specify condition
2. If time is over it will throw Timeout exception
3. Whichever method you wish to synchronized just above that use this

Syntax:

```
WebdriverWait wait=new  
WedDriverWait(driver,10);  
  
wait.until(expectedconditions. );
```



## GetWindowHandle Vs GetWindowHandles

### GetWindowHandle

1. Return type is String
2. It will retrieve the id of the current browser

### GetWindowHandles

1. Return type is set of string
2. It will retrieve the id of all the browser including parent and child.