OOPS and Concepts

Static Keyword:

Static is a keyword used in java with the member function or member variables of a class so that we call call this members without using its object.

- Static Methods does not require any object
- We cannot use a non-static variable in a static method

Variable and Methods can be of two types:

1. Static Member: Need class to call variables and method
2. Non-Static Member: Need Object(Instance) to call variable and method.

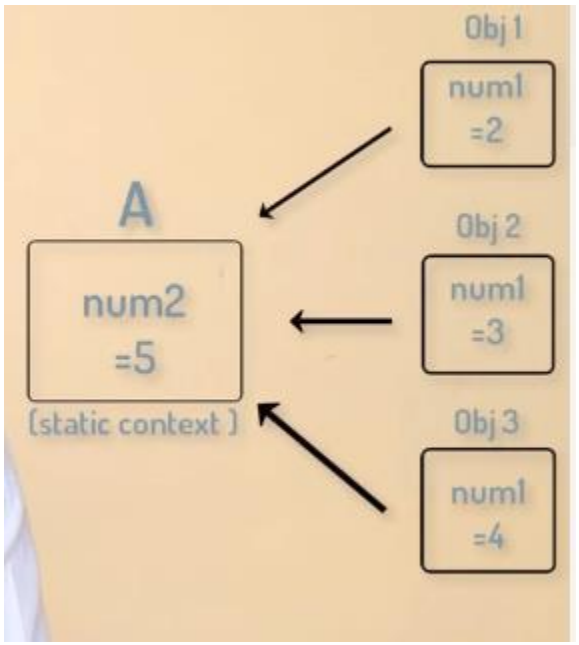
```
public class StaticDemo {

	public static void main(String[] args) {
	A.num2=5;
	A obj1=new A();
	A obj2=new A();
	A obj3=new A();

	obj1.num1=2;
	obj2.num1=3;
	obj3.num1=4;
	}

}

class A{
	int num1;
	static int num2;
	public void show() {
		num2=9;
	}
}
```



All the objects will have same value for **num2** that is 5. we do not have to use the object name here by class name only we can get the value. Where as each object of the instance variable will have its own value

- All the objects of the Static variable will have the same value.
- Every object will have its own value for instance variable.

Q] Can we use a static variable inside a non-static method?
Ans: Yes we can do that.

Here in the above example we can see that the non-static method is trying to access the static variable. It will run because the static variable will be shared between all the methods. That is every object can use the same static object.

Q] Can we use a non-static variable inside a static method?
Ans: No.
Because every object will have its own value and if we try to change the value of the object in static methods we don't know which object value will change. So, this is not possible.

```
public class StaticDemo {

	public static void main(String[] args) {
	A.num2=5;
	A obj1=new A();
	A obj2=new A();
	A obj3=new A();

	A.increment();
	obj1.num1=2;
	obj2.num1=3;
	obj3.num1=4;
	}

}

class A{
	int num1;
	static int num2;
	public static void increment() {
		num1++; //Error
	}
}
```

Count the number of objects of a class in java

```
public class NoOfObjects {
	static int i=0;
	public NoOfObjects(){
		i++;
	}
	public void counter() {
		System.out.println("No of Objects:"+i);
	}
	public static void main(String[] args)
	{
		NoOfObjects n1=new NoOfObjects();
		NoOfObjects n2=new NoOfObjects();
		NoOfObjects n3=new NoOfObjects();
		NoOfObjects n4=new NoOfObjects();
		n4.counter();
	}
}
```

Output :
If i is static variable then ,
No of objects : 4

If i is non-static variable then ,
No of objects : 1

If i is non-static(Instance) variable each time the object is created the value gets initialized to 0.

Static Block

Static Block is a block which is executed when a class is loaded into the JVM, before main method.
To initialize static variable static block is used. Here we cannot use constructors since constructors only work on instance variable.

Static Import:

```
package com.staticdemo;
import static java.lang.System.out; //We have import static method out
import static com.staticdemo.Sample.show; //We have import static method show

public class StaticImportDemo {
	public static void main(String[] args) {
		out.println("Hello how are you");
		show();
	}
}
class Sample{
	public static void show() {

	}
}
```

Q] What is the difference between a Class and an object?

Ans: A class is a model for creating object and does not exist physically

An object is anything that exist physically. Both classes and objects contain Method and Variables.

Q]What is Object Oriented Programming?

Ans: Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods. A feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated.

Q]What are the features of Object-Oriented Programming?

Ans:

The object-oriented programming has the following concepts:

- Inheritance
- Polymorphism
- Abstraction
- Encapsulation
- Association:
  I. Composition
  II. Aggregation

SUPER KEYWORD:

Super keyword is used access the super class methods and properties in the sub class methods. This keyword helps to access the **super** class members without creating any instances.

- Super keyword can only be used in a class with extends another class.
- Super keyword can also be used to invoke the super class constructors. Super(Parameters) should be used in the first line of the constructor.

```
public class Car {

	public static void main(String[] args) {
		Car c=new Suzuki();
	}
}
class Suzuki extends Car{
	public Suzuki() {
		super();
		System.out.println();
	}

}
```

POLYMORPHISM:

Polymorphism