Static class in Java

In java, we have static instance variables as well as static methods and also static block. Classes can also be made static in Java.
Java allows us to define a class within another class. Such a class is called a nested class. The class which enclosed nested class is known as Outer class. In java, we can't make Top level class static. ***Only nested classes can be static***.

What are the differences between static and non-static nested classes?
Following are major differences between static nested class and non-static nested class. Non-static nested class is also called Inner Class.
1) Nested static class doesn't need reference of Outer class, but Non-static nested class or Inner class requires Outer class reference.
2) Inner class(or non-static nested class) can access both static and non-static members of Outer class. A static class cannot access non-static members of the Outer class. It can access only static members of Outer class.
3) An instance of Inner class cannot be created without an instance of outer class and an Inner class can reference data and methods defined in Outer class in which it nests, so we don't need to pass reference of an object to the constructor of the Inner class. For this reason Inner classes can make program simple and concise.

```
/* Java program to demonstrate how to implement static and non-static
classes in a java program. */
class OuterClass{
private static String msg = "CECS 277";

// Static nested class
public static class NestedStaticClass{

        // Only static members of Outer class is directly accessible in nested
        // static class
        public void printMessage() {

                // Try making 'message' a non-static variable, there will be
                // compiler error
                System.out.println("Message from nested static class: " + msg);
        }
        }

        // non-static nested class - also called Inner class
        public class InnerClass{

        // Both static and non-static members of Outer class are accessible in
        // this Inner class
```

```
	public void display(){
		System.out.println("Message from non-static nested class: "+ msg);
	}
	}
}
class Main
{
	// How to create instance of static and non static nested class?
	public static void main(String args[]){

	// create instance of nested Static class
	OuterClass.NestedStaticClass printer = new OuterClass.NestedStaticClass();

	// call non static method of nested static class
	printer.printMessage();

	// In order to create instance of Inner class we need an Outer class
	// instance. Let us create Outer class instance for creating
	// non-static nested class
	OuterClass outer = new OuterClass();
	OuterClass.InnerClass inner = outer.new InnerClass();

	// calling non-static method of Inner class
	inner.display();

	// we can also combine above steps in one step to create instance of
	// Inner class
	OuterClass.InnerClass innerObject = new OuterClass().new InnerClass();

	// similarly we can now call Inner class method
	innerObject.display();
	}
}
```

Output:

```
Message from nested static class: CECS 277
Message from non-static nested class: CECS 277
Message from non-static nested class: CECS 277
```

What is Static Variable in Java?

Static variable in Java is variable which belongs to the class and initialized only once at the start of the execution.

- It is a variable which belongs to the class and not to object(instance)
- Static variables are initialized only once, at the start of the execution. These variables will be initialized first, before the initialization of any instance variables
- A single copy to be shared by all instances of the class
- A static variable can be accessed directly by the class name and doesn't need any object

Syntax :

<class-name>.<variable-name>

What is Static Method in Java?

Static method in Java is a method which belongs to the class and not to the object. A static method can access only static data.

- It is a method which belongs to the class and not to the object(instance)
- A static method can access only static data. It can not access non-static data (instance variables)
- A static method can call only other static methods and can not call a non-static method from it.
- A static method can be accessed directly by the class name and doesn't need any object
- A static method cannot refer to "this" or "super" keywords in anyway

Syntax :

<class-name>.<method-name>

Note: main method is static, since it must be accessible for an application to run, before any instantiation takes place.

ets learn the nuances of the static keywords by doing some excercises!

Example: How to call static variables & methods

Step 1) Copy the following code into a editor

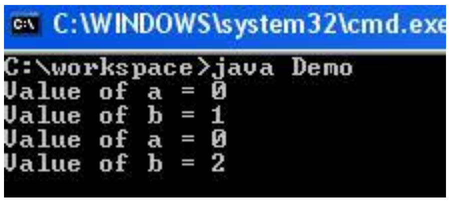
```
public class Demo{
   public static void main(String args[]){
     Student s1 = new Student();
     s1.showData();
     Student s2 = new Student();
     s2.showData();
     //Student.b++;
     //s1.showData();
  }
}

class Student {
int a; //initialized to zero
static int b; //initialized to zero only when class is loaded not for each object
created.

  Student(){
   //Constructor incrementing static variable b
   b++;
  }

   public void showData(){
      System.out.println("Value of a = "+a);
      System.out.println("Value of b = "+b);
   }
//public static void increment(){
//a++;
//}
}
```

<u>Output</u>



```
C:\WINDOWS\system32\cmd.exe
C:\workspace>java Demo
Value of a = 0
Value of b = 1
Value of a = 0
Value of b = 2
```

It is possible to access a static variable from outside the class using the syntax **ClassName.Variable_Name**.

It is not possible to access instance variable "**a**" from java static class method "**increment**".

Java Static Block

The static block is a block of statement inside a Java class that will be executed when a class is first loaded into the JVM

```
class Test{
 static {
 //Code goes here
 }
}
```

A **static block helps to initialize the static data members**, just like constructors help to initialize instance members

Following program is the example of java static block.

Example: How to access static block

```
public class Demo {
 static int a;
 static int b;
 static {
    a = 10;
    b = 20;
 }
 public static void main(String args[]) {

  System.out.println("Value of a = " + a);
  System.out.println("Value of b = " + b);

            }
}
```

you will get following output of the program.

```
Value of a = 10
Value of b = 20
```