Course Name - Object Oriented Programming using Java

Lecture 14 – Static variables & methods. Nested & inner classes

Presented By
Dr. Sudipta Sahana
Asso. Prof.
Dept. of CSE
UEM - Kolkata

Topic of Interest

- Static keyword in Java
- Java static variable
- Java static method
- Nested Classes
- Inner Class

Static keyword in Java

The static keyword in java is used for memory management mainly. We can apply java static keyword with variables, methods, blocks and nested class. The static keyword belongs to the class rather than instance of the class.

The static can be:

- variable (also known as class variable)
- method (also known as class method)
- block
- nested class

Java static variable

The static variable can be used to refer the common property of all objects (that is not unique for each object) e.g. company name of employees, college name of students etc. The static variable gets memory only once in class area at the time of class loading.

.

```
class Student8{
   int rollno;
   String name;
   static String college ="ITS";

   Student8(int r,String n){
   rollno = r;
   name = n;
   }
void display
(){System.out.println(rollno+"
"+name+" "+college);}
```

```
 public static void main(String
args[]){
 Student8 s1 = new
Student8(111,"Karan");
 Student8 s2 = new
Student8(222,"Aryan");

 s1.display();
 s2.display();
 }
}
```

Output -

111 Karan ITS

222 Aryan ITS

Java static method

- A static method belongs to the class rather than object of a class.
- A static method can be invoked without the need for creating an instance of a class.
- static method can access static data member and can change the value of it.

```
class Student9{
    int rollno;
    String name;
    static String college = "ITS";

    static void change(){
    college = “UEM”;
    }

    Student9(int r, String n){
    rollno = r;
    name = n;
    }
```

```
void display
(){System.out.println(rollno+"
"+name+" "+college);}
    public static void main(String
args[]){
    Student9.change();
    Student9 s1 = new Student9
(111,"Karan");
    Student9 s2 = new Student9
(222,"Aryan");
    Student9 s3 = new Student9
(333,“Deep");
    s1.display();
    s2.display();
    s3.display();
    }
}
```

Output -

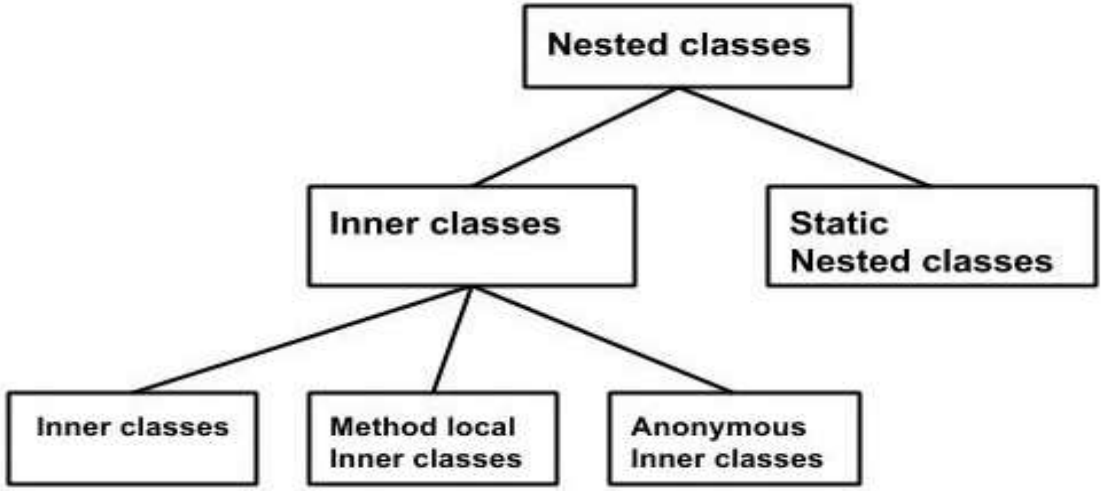
111 Karan UEM
222 Aryan UEM
333 Deep UEM

Nested Classes

In Java, just like methods, variables of a class too can have another class as its member. Writing a class within another is allowed in Java. The class written within is called the nested class, and the class that holds the inner class is called the outer class

Nested classes are divided into two types –

- Non-static nested classes − These are the non-static members of a class.
- Static nested classes − These are the static members of a class.



Inner Class

Creating an inner class is quite simple. You just need to write a class within a class.

```
class Outer {
int outer_x = 100;
void test() {
Inner inner = new Inner();
inner.display();
}
// this is an inner class
class Inner {
void display() {
System.out.println("display: outer_x
= " + outer_x);
}}}
class InnerClassDemo {
public static void main(String args[])
{ Outer outer = new Outer();
outer.test();
} }
```

Output-

display: outer_x = 100

Note - It is important to realize that class **Inner** is known only within the scope of class **Outer**. The Java compiler generates an error message if any code outside of class **Outer** attempts to instantiate class **Inner**. Generalizing, a nested class is no different than any other program element: it is known only within its enclosing scope.

An inner class has access to all of the members of its enclosing class, but the reverse is not true. Members of the inner class are known only within the scope of the inner class and may not be used by the outer class

Thank You