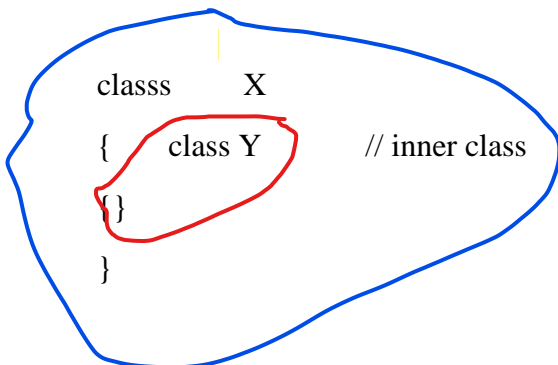


Inner and Nested class

There are two types of nested classes you can create in Java.

- Non-static nested class (inner class)
- Static nested class



```
Class X
{ static class Y // nested class
{}
}
```

// Demonstrate an inner 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 Inner_Demo
```

```

{
public static void main(String args[]) {
    Outer outer = new Outer();
    outer.test();
}
}
O/p : display: outer_x =100
class Outer {
    void outerMethod() {
        System.out.println("Inside .....outerMethod");
        // Inner class is local to outerMethod()
        class Inner {
            void innerMethod() {
                System.out.println("Inside innerMethod.....");
            }
        }
        Inner y = new Inner();
        y.innerMethod();
    }
}
class InnerClassTest
{
    public static void main(String[] args)
    {
        Outer x = new Outer();
        x.outerMethod();
    }
}

```

O/P: InsideouterMethod

Inside innerMethod.....

static nested class

```
class N1
{
    // static class ie nested class
    static class N2
    {
        int n1 = 20;
        int n2 = 30;
        int getTotal()
        {
            return n1 + n2;
        }
    }
}

public class NestedTest
{
    public static void main(String[] args) {

        // create an object of the static nested class
        // using the name of the outer class
        N1.N2 obj = new N1.N2();
        System.out.println("Total == " + obj.getTotal());
    }
}
```

O/P: Total == 50

Key Points to Remember

- Java treats the inner class as a regular member of a class. They are just like methods and variables declared inside a class.
- Since inner classes are members of the outer class, you can apply any access modifiers like private, protected to your inner class which is not possible in normal classes.
- Since the nested class is a member of its enclosing outer class, you can use the dot (.) notation to access the nested class and its members.
- Using the nested class will make your code more readable and provide better encapsulation.
- Non-static nested classes (**inner classes**) have access to other members of the outer/enclosing class, even if they are declared private.

```
// nested class
class Outer1
{ static int x=20;
  static class Inner1
  {
    void method ()
    {System.out.println("nested class method executed: and x== "+x);}
  }
}

class NestedDemo
{ public static void main(String a[])
  { Outer1.Inner1 inobj=new Outer1.Inner1(); // object of inner class
    inobj.method();
  }
}
```

O/P :nested class method executed: and x==20

Note: if Inner class is not static then above program will give compile time error and to solve this problem, we have to create object of outer class as shown below

```
// Inner class
class Outer2
{ int x=20;
  class Inner2
  {
    void method ()
    {System.out.println("inner class method executed: and x== "+x);}
  }
}

class InnerDemo
{ public static void main(String a[])
  { Outer2 outobj= new Outer2();
    Outer2.Inner2 inobj=outobj.new Inner2();
    inobj.method();
  }
}
```

O/P : inner class method executed: and x==20