Inner Class in Java

A class declared inside another class is known as **nested classes in java**. The scope of a nested class is tied by the scope of its enclosing class (outer class).

Thus,if class B is defined inside class A, then class B cannot exist independently without class A. Here, class A is outer or enclosing class and class B is nested class.

Inner class in Java

An inner class in java is a class that is declared inside of another class without static modifier. It is also commonly known as a non-static nested class in Java. It can access all members (variables and methods) of its outer class.

An inner class cannot have any kind of static members. The members of the java inner class may be:

1. Instance variables
2. Instance methods
3. Constructors
4. Initializer block
5. Inner class

```
package innerClass;
public class A
{
// A member inner class named B.
  class B
  {
   public void m1()
   {
    System.out.println("Inner class method");
   }
  }
public static void main(String[] args)
{
// Static area of the outer class.
```

```
   System.out.println("Outer class main method");

// Create an instance of class A.
     //A a = new A();
// Create an instance of class B and call method m1() from the static area of outer cl
ass using reference variable i.
    // A.B b = a.new B();
      //b.m1();
   }
}
```

```
package innerClass;
public class A
{
 class B
 {
  public void m1() {
    System.out.println("Inner class method");
 }
}




// An instance area of outer class.
   void m2() { // Instance method of outer class.
     System.out.println("Outer class instance method");

// To call method m1 of an inner class B, here we create directly an object of inner c
lass in the instance area.
    B b = new B();
    b.m1();








  }
public static void main(String[] args)
{
// Create an instance of class A.
     A a = new A();
      a.m2();
  }
}
```

```
package innerClass;
public class MyOuter2
```

```
{
// Declare an instance variable of the outer class and initialize the value 20.
    int x = 20;
// A regular member inner class start here.
    class MyInner2 {
  // An instance variable of member inner class.
       int x = 30;
     public void showValue() {
     // A local variable of member inner class.
         int x = 40;
         System.out.println(x); // It will print the value 40 of local variable x.
         System.out.println(this.x); // It will print the value 30 of instance variabl
e of member inner class because here this refers to the current inner class object.
         System.out.println(MyOuter2.this.x); // It will print the value 20 of instanc
e variable of outer class because here this refers to current outer class object.
       }
     }
public static void main(String[] args) {
     MyOuter2 mo = new MyOuter2();
     MyOuter2.MyInner2 mi = mo.new MyInner2();
      mi.showValue();
    }
}
```

Method Local Inner Class in Java with Example

An inner class that is declared inside a method of the outer class is called **method local inner class in Java**.

Its scope is limited to the block of a method in which it is declared.

Therefore, the declaration of method local inner class cannot use any access modifiers such as public, protected, private, and non-access modifiers such as static.

We can also declare method local inner class in Java inside the constructor, static initializers, and non-static initializers.

```
package innerClass;
public class OuterClass
{
// An instance method of class OuterClass.
   public void display()
  {
// Declaration of a method local inner class.
    class InnerClass
    {
       public void msg(){
        System.out.println("I am in Local inner class");
       }
```

```
     }
// Now, create an instance of a method local inner class and call the msg() method using object reference variable ic.
     InnerClass ic = new InnerClass();
       ic.msg();
  }
public static void main(String[] args)
{
// Create an object of the outer class OuterClass.
     OuterClass oc = new OuterClass();
       oc.display();
   }
}
```

```
package innerClass;
public class Test1
{
  public void m1()
  {
    void sum(int x, int y)
    {
     System.out.println("Sum of two numbers: " +(x+y));
    }
   }
public static void main(String[] args)
{
   Test1 t = new Test1();
    t.m1();
    t.sum(20,30);
    t.sum(10,40);
    t.sum(50,50);
    t.sum(100,100);
  }
}
```