**Nested Class**

package com.nested;

/\*

//Using Member Class

class A

{

void display()

{

B ob=new B();

ob.msg();

}

class B

{

void msg()

{

System.out.println("Inside B");

}

}

}

public class TestNestedClass

{

public static void main(String[] args)

{

A a=new A();

a.display();

}

}

\*/

//Using local Inner class

/\*

class Outer

{

void display()

{

class Inner

{

void msg()

{

System.out.println("Inside display method:");

}

}

Inner ob=new Inner();

ob.msg();

}

}

public class TestNestedClass

{

public static void main(String[] args)

{

Outer a=new Outer();

a.display();

}

}

\*/

/\*

//Using Anonymous class

abstract class Vehicle

{

abstract void details();

}

public class TestNestedClass

{

public static void main(String[] args)

{

Vehicle v=new Vehicle()

{

void details()

{

System.out.println("Company Maruti:");

}

};

v.details();

}

}

\*/

/\*

interface Vehicle

{

abstract void details();

}

public class TestNestedClass

{

public static void main(String[] args)

{

Vehicle v=new Vehicle()

{

public void details()

{

System.out.println("Company Maruti:");

}

};

v.details();

}

}

\*/

/\*

//using thread

public class TestNestedClass

{

public static void main(String[] args)

{

Thread v=new Thread()

{

public void run()

{

System.out.println("Create a Thread:");

}

};

v.start();

}

}\*/

//using runnable

/\*

public class TestNestedClass

{

public static void main(String[] args)

{

Runnable v=new Runnable()

{

public void run()

{

System.out.println("Create a Thread:");

}

};

new Thread().start();;

}

}

\*/

//using static

class Outer

{

static int data=20;

static class Inner

{

int y=10;

void msg()

{

System.out.println(data+y);

}

}

}

public class TestNestedClass

{

public static void main(String[] args)

{

Outer.Inner ob=new Outer.Inner();

ob.msg();

}

}