Inheritance

class A {

public void display() {

System.out.println("Class A");

}}

class B extends A {

public void show() {

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

}}

public class Main {

public static void main(String[] args) {

A obj1 = new A();

obj1.display();

B obj2 = new B();

obj2.display();

obj2.show(); }}

Output

Class A

Class A

Class B

Interface

interface P {

void p1();

}

class A implements P {

public void p1() {

System.out.println("Hello");

}}

public class Main {

public static void main(String[] args) {

A obj = new A();

obj.p1(); } }

Output:

Hello

Constructor

class A {

A() {

System.out.println("I am in A");

}

}

public class Main {

public static void main(String[] args) {

A b = new A();

}

}

Output

I am in A

Method

class M {

void M() {

System.out.println("This is my method.");

}

}

public class Main {

public static void main(String[] args) {

M obj = new M();

obj.M();

}

}

OUTPUT

This is my method.

Static

class M {

static int count;

static void M() {

System.out.println("This is a static method.");

}

static {

System.out.println("Static initializer block");

}

static class InnerClass {

void display() {

System.out.println("Static nested class");

}}

public static void main(String[] args) {

count = 5;

System.out.println(count);

M();

M.InnerClass obj = new M.InnerClass();

obj.display();

}}

Output:

5

This is a static method.

Static initializer block

Static nested class