1) IIB:-

1)Instant initialization block IIB are executed when object are created

2) Number of time we create object same number of time IIB will be called

3) IIB are used to initialized all the instants variable in one place and that gives us better redability of the code

Example 1:-

**package** practies;

**public** **class** Demo{

{

System.***out***.println("from IIB");

}

**public** **static** **void** main(String[] args) {

}

}

OUTPUT:-no output reason will be not create object then output no

Example 2:-

**package** practies;

**public** **class** Demo{

{

System.***out***.println("from IIB");

}

**public** **static** **void** main(String[] args) {

Demo d1=new Demo();

}

}

OUTPUT:-From IIB

Example 3:-

**package** practies;

**public** **class** Demo{

{

System.***out***.println("from IIB");

}

**public** **static** **void** main(String[] args) {

Demo d1=new Demo();

Demo d2=new Demo();

}

}

OUTPUT:-From IIB

From IIB

Example 4:-

**package** practies;

**public** **class** Demo{

{

System.***out***.println("from IIB");

}

Demo(){

System.***out***.println("From Constructor");

}

**public** **static** **void** main(String[] args) {

Demo d1=**new** Demo();

// Demo d2=new Demo();

}

}

OUTPUT:-From IIB

From Constructor

Example 5:-

**package** practies;

**public** **class** Demo{

{

System.***out***.println("from IIB");

}

Demo(){

System.***out***.println("From Constructor");

}

**public** **static** **void** main(String[] args) {

Demo d1=**new** Demo();

System.***out***.println("From Main");

}

}

OUTPUT:-from IIB

From Constructor

From Main

Example 6:-

**package** practies;

**public** **class** Demo{

{

System.***out***.println("from IIB");

}

Demo(){

System.***out***.println("From Constructor");

}

**public** **static** **void** main(String[] args) {

System.***out***.println("start Main");

Demo d1=**new** Demo();

System.***out***.println("End Main");

}

}

OUTPUT:- start Main

from IIB

From Constructor

End Main

Example 7:-

**package** practies;

**public** **class** Demo{

Demo(){

System.***out***.println("From Constructor");

}

{

System.***out***.println("from IIB-2");//always first call IIB series

}

{

System.***out***.println("from IIB-1");

}

**public** **static** **void** main(String[] args) {

Demo d1=**new** Demo();

}

}

OUTPUT :- from IIB-2

from IIB-1

From Constructor

We can initialize both static and non-static variables inside IIB.

Example 8

**package** practies;

**public** **class** Demo{

**static** **int** *i*;

{

*i*=20;

System.***out***.println(*i*);

}

**public** **static** **void** main(String[] args) {

Demo d1=**new** Demo();

}

}

OUTPUT :- 20

2) SIB:-

SIB run before main method and does not required any invoking statement.

Example 1:-

**package** practies;

**public** **class** Demo{

**static**

{

System.***out***.println("From SIB");

}

**public** **static** **void** main(String[] args) {

}

}

OUTPUT :-

From SIB

Example 2:-

**package** practies;

**public** **class** Demo{

**static**

{

System.***out***.println("From SIB");

}

**public** **static** **void** main(String[] args) {

System.***out***.println("From Main");

}

}

OUTPUT :-

From SIB

From Main

Example 3:-

**package** practies;

**public** **class** Demo{

**static** **int** *i*;

**static**

{

*i*=200;

System.***out***.println(*i*);

}

**public** **static** **void** main(String[] args) {

System.***out***.println("From Main");

}

}

OUTPUT :-

200

From Main

Example 4:-

**package** practies;

**public** **class** Demo{

**static**

{

System.***out***.println("From SIB-2");

}

**static**

{

System.***out***.println("From SIB-1");

}

**public** **static** **void** main(String[] args) {

System.***out***.println("From Main");

}

}

OUTPUT :-

From SIB-2

From SIB-1

From Main

We can not initialize non static variable inside SIB

Example 5:-

**package** practies;

**public** **class** Demo{

**int** i;

**static**

{

i=200; //Error

}

**public** **static** **void** main(String[] args) {

System.***out***.println("From Main");

}

}

OUTPUT:-

Error We can not initialize non static variable inside SIB

Session 12:-

Example1:-

**package** practies;

**public** **class** Demo{

{

System.***out***.println("IIB");

}

**static**

{

System.***out***.println("SIB");

}

**public** **static** **void** main(String[] args) {

Demo d1=**new** Demo();

}

}

OUTPUT:-

SIB

IIB

Example 2 :-

**package** practies;

**public** **class** Demo{

{

System.***out***.println("IIB");

}

**static**

{

System.***out***.println("SIB");

}

Demo(){

System.***out***.println("Demo");

}

**public** **static** **void** main(String[] args) {

Demo d1=**new** Demo();

}

}

OUTPUT:- SIB

IIB

Demo

Example 3:-

package practies;

public class Demo {

{

System.out.println("IIB");

}

{

System.out.println("IIB-2");

}

static {

System.out.println("SIB");

}

Demo(){

System.out.println("Demo");

}

public static void main(String[] args) {

Demo d1=new Demo();

System.out.println("main");

}

}

OUTPUT:-

SIB

IIB

IIB-2

Demo

main

Example 4:-

package practies;

public class Demo {

{

System.out.println("IIB");

}

static

{

new Demo();

System.out.println("SIB");

}

Demo(){

System.out.println("Demo");

}

public static void main(String[] args) {

System.out.println("Main");

}

}

OUTPUT :-

IIB

Demo

SIB

Main

Exmple 5:-

package practies;

public class Demo {

static {

System.out.println("SIB");

}

static {

System.out.println("HEllo");

}

{

System.out.println("IIB-");

}

public static void main(String[] args) {

new Demo();

}

}

OUTPUT:-

SIB

HEllo

IIB-

Example 7:-

package practies;

public class Demo {

public void test()

{

System.out.println("Test");

}

public static void main(String[] args) {

new Demo().test();

}

{

System.out.println("IIB");

}

Demo(){

System.out.println("Demo");

}

}

Output:-

IIB

Demo

Test

IF you create an object inside IIB you will get an error but then the program will hard updraft.

Example 8:-

package practies;

public class Demo{

{

new Demo();

}

public static void main(String[] args) {

Demo d1=new Demo();

System.out.println("From main");

}

}

OUtput:-Exception in thread "main" java.lang.StackOverflowError

at practies.Demo.<init>(Demo.java:6)