Static Blocks & Instance Blocks

JAVA CLASS ELEMENTS

Class Test

{

i) Variables

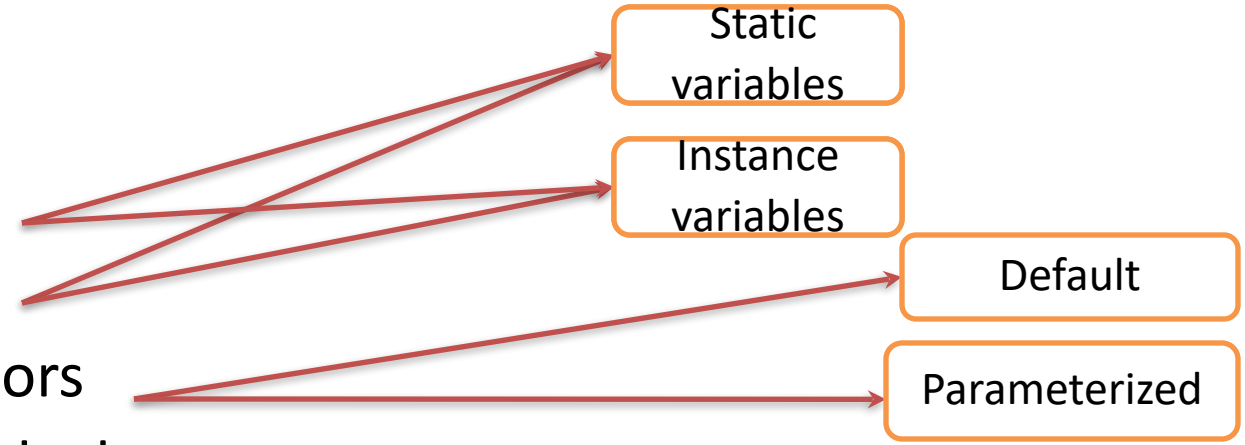
ii) Methods

iii) Constructors

iv) Instance blocks

v) Static blocks

}



Instance Blocks in JAVA

```
class Test
{
    Test()
    {
        System.out.println("0-arg");
    }
    {
        System.out.println("instance block");
    }
    public static void main()
    {
        Test t = new Test();
    }
}
```

Constructor

Instance block

Output:
instance block
0-arg

```
{
    System.out.println("instance block");
}
```

Syntax

```
{
    //logics here
}
```

Both the constructors and instance blocks are executed during object creation

During object creation,

1. Instance blocks
2. Constructors are executed.

Need of instance blocks over constructors!

```
public class Test
{
    Test()
    {
        System.out.println("0-arg");
    }
    Test(int a)
    {
        System.out.println("1-arg");
    }
    {
        System.out.println("instance block");
    }
    public static void main(String args[])
    {
        Test t = new Test();
        new Test(10);
    }
}
```

01

02

Constructors are specific to objects

Common to all objects

Output:

instance block

0-arg

instance block

1-arg

```java
public class Test
{
    Test()
    {
        this(10);
        System.out.println("0-arg");
    }
    Test(int a)
    {
        System.out.println("1-arg");
    }
    {
        System.out.println("instance block - 1");
    }
    {
        System.out.println("instance block - 2");
    }
    public static void main(String args[])
    {
        Test t = new Test();
    }
}
```

Output:
instance block – 1
instance block – 2
1-arg
0-arg

1. There can be multiple instance blocks.
2. The order of execution is from top to bottom
3. The instance blocks depends upon object creation and not on constructor calling.

Static blocks in JAVA

```
class Test
{
    Test()
    {
        System.out.println("0-arg");
    }
    static
    {
       System.out.println("static block");
    }
    {
        System.out.println("instance block");
    }
    public static void main()
    {
        Test t = new Test();
    }
}
```

Constructor

Static block

Instance block

Output:
Static block
Instance block
Constructor

```
static
{
    System.out.println("static block");
}
```

Syntax:

```
static
{
    //logics here
}
```

Use:

To write the logics which are executed during .class file loading.

.class file is loaded only once.

Static blocks are executed only once.

```
public class Test
{
    Test()
    {
        System.out.println("0-arg");
    }
    {
        System.out.println("instance block");
    }
    static
    {
        System.out.println("static block - 1");
    }
    static
    {
        System.out.println("static block - 2");
    }
    public static void main(String args[])
    {
        Test t = new Test();
    }
}
```

1. There can be multiple static blocks.
2. The order of execution is from top to bottom
3. When the program contains static, instance and constructor, the priority is

1. static
2. instance
3. constructors

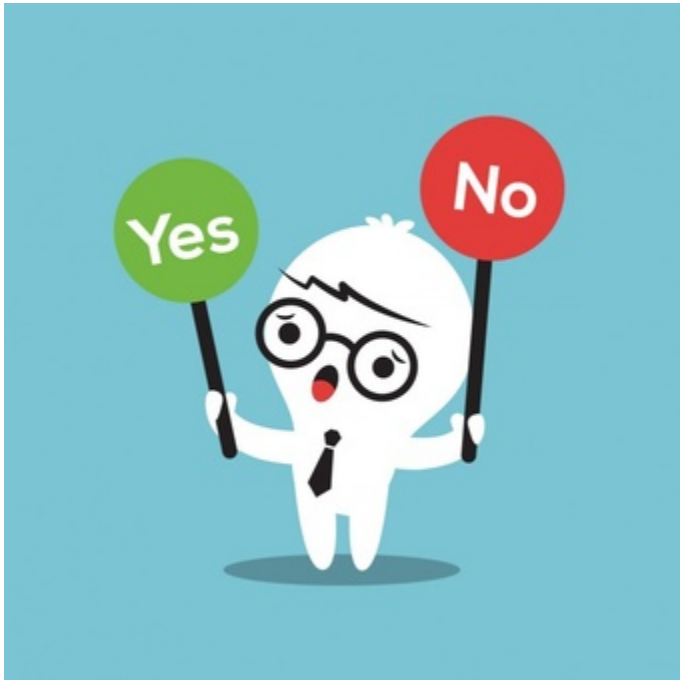
Output:
static block – 1
static block – 2
instance block
0-arg

Static blocks are executed during .class file loading, therefore even if no object is created, static blocks are loaded.

```
public class Test
{
    static
    {
        System.out.println("static block - 1");
    }
    static
    {
        System.out.println("static block - 2");
    }
    public static void main(String args[])
    {
    }
}
```

Output:
static block – 1
static block - 2

Is it possible to execute static blocks without main?



Both are correct!!

If the java version is 1 or 1.5, execution of static blocks does not need main method

If the java version is above 1.5, main method is mandatory.

```
import java.lang.*;
public class A
{
    static
    {
        System.out.println(" A- class static block");
    }
    public static void main(String args[])
    {
        try
        {
            Class.forName("B");
        }
        catch(ClassNotFoundException ex)
        {
            System.out.println(ex.toString());
        }
    }
}
class B
{ static
    {       System.out.println(" B- class static block");
    }
}
```

There are two static blocks in the given program.

To execute the static blocks, main file is mandatory.
But only one main is possible!!

Static blocks are used to initialize static variables

```
public class Test
{
    static int eid;
    static
    {
        eid = 222;
    }
    static void disp()
    {
        System.out.println(Test.eid);
    }
    public static void main(String args[])
    {
        Test.disp();
    }
}
```

Output:
222

```java
public class Test
{
    int a = 10;
    static int b =20;
    void m1(int a)
    {
        System.out.println("instance method");
    }
    static void m2(String str)
    {
       System.out.println("static method");
    }
    {
        System.out.println("instance block");
    }
    static
    {
        System.out.println("static block");
    }

    Test()
    {
        System.out.println("0-arg");
    }
    Test(int a)
    {
        System.out.println("1-arg");
    }
    public static void main(String args[])
    {
        Test t = new Test();
        new Test(10);
        t.m1(10);
        Test.m2("face");
    }
}
```

Output?

static block
instance block
0-arg
instance block
1-arg
instance method
static method

Predict the Output?

```
class Test
{
    public static void main
   (String[] args)
    {
        int a =5,b;
        a++;
        b = a;
        System.out.println(a+"
   "+b);
    }
}
```

Output ?

Why?

6 6

With static

```
class Test
{
   public static void main (String[]
   args)
    {
        int a =5,b;
        a++;
        b = a;
        System.out.println(b);
    }
```

```
    static
     {
         int a=5;
         System.out.println(a);
     }
}
```

Output ?

6

How?

MCQ

```
class Test1
{
public
    static void main(String[] args)
    {
        int x = 20;
        System.out.println(x);
    }
    static
    {
        int x = 10;
        System.out.print(x + " ");
    }
}
```

A) 10 20 R

B) 20 10

C) 10 10

D) 20 20

```
class Test1 {
    int x = 10;
public
    static void main(String[] args)
    {
        System.out.println(x);
    }
    static
    {
        System.out.print(x + " ");
    }
}
```

A) 10 10

B) Error R

C) Exception

D) none

```
class Test1
{
    int x = 10;
public
    static void main(String[] args)
    {
        Test1 t1 = new Test1();
        System.out.println(t1.x);
    }
    static
    {
        int x = 20;
        System.out.print(x + " ");
    }
}
```

A) 10 20

B) 20 10 R

C) 10 10

D) Error

```
class Test1
{
    int x = 10;
public
    static void main(String[] args)
    {
        System.out.println(Test1.x)
   ;
    }
    static
    {
        int x = 20;
        System.out.print(x + " ");
    }
}
```

A)10 10

B) 20 20

C) 20 10

D) Error R


Thank you