

Java OOPS V.V.I

Date : / /

Q. What is OOP? full explanation.

Ans → OOP Stands for Object Oriented programming language, the main purpose of OOP is to deal with real world entity using programming language.

OOPS Features :-

- class
- object

SUBSCRIBE

using programming language.

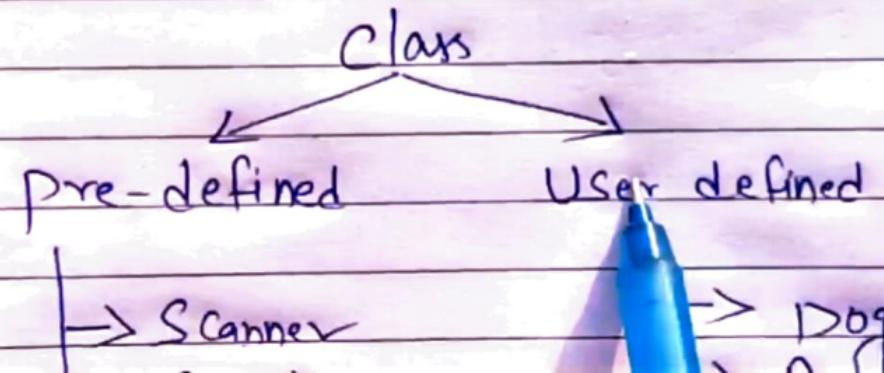
OOPS Features :-

- class
- Object
- Inheritance
- Polymorphism
- Encapsulation
- Abstraction.

Class and Object

Q. What is class ?

Ans -> class is a collection of objects and it doesn't take any space on memory,
Class is also called as blueprint / logical entity.



Class

Pre-defined

User defined

→ Scanner

→ Console

→ System

→ String

→ Dog

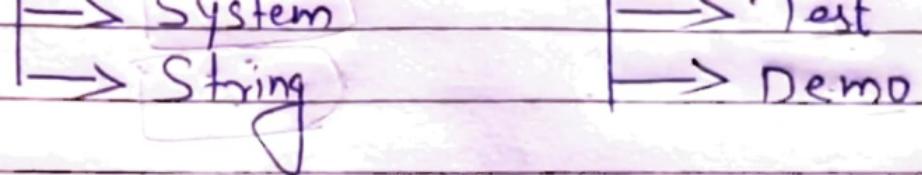
→ A

→ Test

→ Demo

User-defined class:- A class which is
created by java programmer
is called user-defined class.

SUBSCRIBE



User-defined class:- A class which is created by java programmer is called user-defined class.

Ex-> Class CLASS-name
 |
 |= = = = // data

 |= = = = // methods

Object:- Object is an instance of class
that executes the class. Once the Object
is created , it takes up space like other
variable in memory.

Syntax:- class-
 ↓
 class name obj-name = new class-name()
 ↓ ↓
 DIMM Constructor

that executes the class. Once the Object is created , it takes up space like other variable in memory.

Syntax:- class-name obj-name = new class-name()
 | | | |
 class name object DMA Constructor
 reference

untitled

```
/* Class & Object */  
class Demo  
{  
    int a=10; String b="ankush";  
  
    void Show()  
    {  
        System.out.print(a+" "+b);  
    }  
}  
class Test  
{  
    Demo r=new Demo();  
    r.Show();
```



```
Test.java  X  /* Class & Object */  
  
class Demo  
{  
    int a=10;  String b="ankush";  
  
    void Show()  
    {  
        System.out.print(a+" "+b);  
    }  
}  
  
class Test  
{  
    public static void main(String[] args) {  
        Demo d = new Demo();  
        d.Show();  
    }  
}
```

Command Prompt

```
C:\Users\WIN10\Desktop>javac Test.j  
C:\Users\WIN10\Desktop>java Test  
10 ankush  
C:\Users\WIN10\Desktop>
```

```
Test.java x

}

    System.out.print(a+" "+b);
}

}

class Test
{
    public static void main(String[] args)
    {
        Demo r;
        r=new Demo();
        r.Show();
    }
}
```

Command Prompt

C:\Users\WIN10\Desktop>javac Test.j

C:\Users\WIN10\Desktop>java Test

C:\Users\WIN10\Desktop>javac Test.j

C:\Users\WIN10\Desktop>java Test

10 ankush

C:\Users\WIN10\Desktop>

Q. What is Constructor? full explanation.

Ans → Constructor is a special type of method whose name is same as class name.

Note :- i) The main purpose of Constructor is initialize the object.

ii) Every Java class has a constructor.

iii) A Constructor is automatically called at the time of object creation.

ii> Every java class has a constructor.

iii> A Constructor is automatically called at the time of object creation.

iv> A Constructor never contain any return-type including void.

Syntax:-

class class-name

{
 class-name()

return-type including void.

Syntax:- class class-name
 {
 class-name()
 {
 }
 }

```
1 // java constructor
2 class A
3 {
4     int a; String name;
5     A()
6     {
7         a=0; name=null;
8     }
9     void show()
10    {
11        System.out.print(a+" "+name);
12    }
13 }
14 
```

```
B.java
4 int a; String name;
5 A()
6 {
7     a=0; name=null;
8 }
9 void show()
10 {
11     System.out.print(a+" "+name);
12 }
13 }
14 class B
15 {
16     public static void main(String[] args)
17     {
18         A ref=new A();
19         ref.show();
20     }
}
```

Select Command Prompt

C:\Users\WIN10\Desktop>javac B.java

```
C:\Users\WIN10\Desktop>java B
0 null
C:\Users\WIN10\Desktop>
```

```
1          /* Java Constructor */
2 class A
3 {
4     int a;  String name;
5     /*A()
6     {
7         a=0; name=null;
8     }*/
9     void show()
10    {
11        System.out.print(a+" "+name);
12    }
13 }
14 class B
15 {
16     public static void main(String[] args) {
17 }
```

Select Command Prompt

C:\Users\WIN10\Desktop>javac B.java

C:\Users\WIN10\Desktop>java B
0 null

C:\Users\WIN10\Desktop>javac B.java

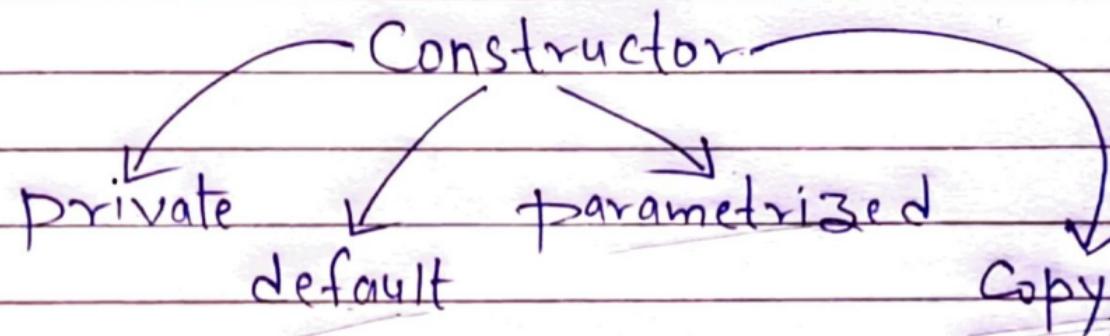
C:\Users\WIN10\Desktop>java B
0 null

C:\Users\WIN10\Desktop>

V.V.I

Constructor

Types of Constructor :-



1. default Constructor

Q. What is default Constructor?

Ans → A Constructor which does not have any parameter is called default Constructor.

Syntax :- class A

A(), No any parameter

Syntax:- class A

S ↗ No any parameter
 |
 A()
 |
 S

{
 }
 {

Note :-

enc

class A

SUBSCRIBE

Note:-



SUBSCRIBE

```
1      /* Default Construcor */
2 class A
3 {
4     int a; String b; boolean c;
5     A()//default
6     {
7         a=100; b="ankit"; c=true;
8     }
9     void Disp()
10    {
11        System.out.print(a+" "+b+" "+c);
12    }
13 } 1
14 class |
```

```
9     void Disp()
10    {
11        System.out.print(a+" "+b+" "+c);
12    }
13 }
14 class B
15 {
16     public static void main(String[] args)
17     {
18         A r=new A();
19         r.Disp();
20     }
21 }
```

Command Prompt

C:\Users\WIN10\Desktop>



```
B.java
9 void Disp()
10 {
11     System.out.print(a+" "+b+" "+c);
12 }
13 }
14 class B
15 {
16     public static void main(String[] args) {
17         A r=new A();
18         rDisp();
19     }
20 }
21 }
```

Command Prompt

C:\Users\WIN10\Desktop>javac B

C:\Users\WIN10\Desktop>java B
100 ankit true
C:\Users\WIN10\Desktop>



```
B.java
1  class A {
2      /* Default Constructor */
3      int a; String b; boolean c;
4      /*A()//default
5      {
6          a=1020; b="ankush"; c=false;
7      }*/
8      void Disp()
9      {
10         System.out.print(a+" "+b+" "+c);
11     }
12 }
13 }
14 class B
15 {
16     public static void main(String[] args) {
17         A r=new A();
```

Line 8, Column 8

Select Command Prompt
'javacB.java' is not recognized as an internal or external command,
operable program or batch file

C:\Users\WIN10\Desktop>javac B

C:\Users\WIN10\Desktop>java B
1020 ankush false

C:\Users\WIN10\Desktop>javac B

C:\Users\WIN10\Desktop>java B
0 null false

C:\Users\WIN10\Desktop>

Tab Size: 4 Java 07:33 03-02-2021



Type here to search



2. Parametrized Constructor

Q. What is parametrized constructor?

Ans → A constructor through which we can pass one or more parameters is called parametrized constructor.

Syntax:— class A

is called parametrized constructor.

Syntax:- class A

A(int x, string y)

}

```
1          /* Parameterized Constructor */  
2 class A  
3 {  
4     A(int a,int b)//parameterized  
5     {  
6     }  
7     void show()  
8     {  
9     }  
10    }  
11 }  
12 class B  
13 {  
14     public static void main(String[] args) {  
15         |  
16     }  
17 }
```

B.java

```
1          /* Parameterized Constructor */  
2 class A  
3 {  
4     int x,y;  
5     A(int a,int b)//parameterized  
6     {  
7         x=a; y=b;  
8     }  
9     A(int a,String b)//parameterized  
10    {  
11        x=a; y=b;  
12    }  
13    void show()  
14    {  
15        System.out.print(x+" "+y);  
16    }  
17 }
```

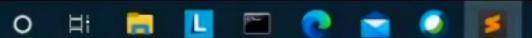
Line 9, Column 20

Tab Size: 4

Java



Type here to search



8:37 03-02-2021

B.java

```
4 int x,y;
5 A(int a,int b)//parameterized
6 {
7     x=a; y=b;
8 }
9 A(int a,String b)//parameterized
10 {
11     System.out.print(a+" "+b);
12 }
13 void show()
14 {
15     System.out.print(x+" "+y);
16 }
17 }
18 class B
19 {
20     public static void main(String[] args) {
```

B.java

```
10  {
11      System.out.print(a+" "+b);
12  }
13  void show()
14  {
15      System.out.print(x+" "+y);
16  }
17 }
18 class B
19 {
20     public static void main(String[] args) {
21         A r=new A(100,200);
22         A ref=new A(10,"ankush");
23         r.show();
24     }
25 }
```



```
B.java
1  /* Parameterized Constructor */
2  class A
3  {
4      int x,y;
5      A(int a,int b)//parameterized
6      {
7          x=a; y=b;
8      }
9      A(int a,String b)//parameterized
10     {
11         System.out.println(a+" "+b);
12     }
13     void show()
14     {
15         System.out.println(x+" "+y);
16     }
17 }
18 class B
19 {
20     public static void main(String[] args) {
21         A r=new A(100,200);
22         r.show();
23         A ref=new A(10,"ankush");
24     }
25 }
```

Select Command Prompt

```
C:\Users\WIN10\Desktop>java B
100 200
C:\Users\WIN10\Desktop>javac B.java
```

```
C:\Users\WIN10\Desktop>java B
10 ankush100 200
C:\Users\WIN10\Desktop>javac B.java
```

```
C:\Users\WIN10\Desktop>java B
100 200
10 ankush
```

```
C:\Users\WIN10\Desktop>
```

3. Copy Constructor

Q. what is Copy Constructor? full detail.

Ans → Whenever we pass object reference to the constructor then it is called copy constructor.

Syntax:- class class-name
 {

 class-name (obj ref)

Ans → Whenever we pass object reference to the constructor then it is called copy constructor.

Syntax:— class class-name

{

class-name(obj ref)

{

}

}

```
1          /* Copy Constructor */
2 class A
3 {
4     int a; String b;
5     A()
6     {
7         a=10;  b="LearnCoding";
8         System.out.println(a+" "+b);
9     }
10    A(A ref)
11    {
12        a=ref.a;
13        b=ref.b;|
14    }
15 }
```

```
4 int a; String b;
5 A()
6 {
7     a=10; b="LearnCoding";
8     System.out.println(a+" "+b);
9 }
10 A(A ref)
11 {
12     a=ref.a;
13     b=ref.b;
14     System.out.println(a+" "+b);
15 }
16 }
17 class |
```



```
B.java
1      /* Copy Constructor */
2  class A
3  {
4      int a; String b;
5      A()
6      {
7          a=10; b="LearnCoding";
8          System.out.println(a+" "+b);
9      }
10     A(A ref)
11     {
12         a=ref.a;
13         b=ref.b;
14         System.out.println(a+" "+b);
15     }
16 }
17 class B
18 {
19     public static void main(String[] args) {
20         A r=new A();
21     }
}
```

Select Command Prompt

C:\Users\WIN10\Desktop>javac B.java

C:\Users\WIN10\Desktop>java B
10 LearnCoding
10 LearnCoding

C:\Users\WIN10\Desktop>



4.- Private Constructor

Q. what is private constructor ?

Ans → In Java, it is possible to write a constructor as a private but according to the rule we can't access private members outside of class .

Syntax :- class class-name
 { }

constructor as a private but according to the rule we can't access private members outside of class.

Syntax :- class class-name
 {

 private class-name()

```
1      /* Private Constructor */
2  class A
3  {
4      int a; double b; String c;
5      private A()
6      {
7          a=10; b=30.56; c="ankit";
8          System.out.println(a+" "+c);
9      }
10     public static void main(String[] args)
11         A r=new A();
12     }
13 }
```

Windows Command Prompt

C:\Users\WIN10\Desktop>javac A.java

```
C:\Users\WIN10\Desktop>java A  
10 ankit
```

C:\Users\WIN10\Desktop>

B.java

```
1  /* Private Constructor */  
2  class A  
3 {  
4      int a; double b; String c;  
5      private A()  
6      {  
7          a=10; b=30.56;  
8          System.out.println("A");  
9      }  
10 }  
11 class B  
12 {  
13     public static void main(String args[]){  
14         A r=new A();  
15     }  
16 }  
17 }
```

Select Command Prompt

```
C:\Users\WIN10\Desktop>javac A.java  
  
C:\Users\WIN10\Desktop>java A  
10 30.56 ankit  
  
C:\Users\WIN10\Desktop>javac B.java  
B.java:15: error: A() has private access in A  
        A r=new A();  
                  ^  
1 error
```

2 characters selected



```
B.java
1          /* Private Constructor */
2 class A
3 {
4     int a; double b; String c;
5     private A()
6     {
7         a=10; b=30.56; c="ankit";
8         System.out.println(a+" "+b+" "+c);
9     }
10    static void Show()
11    {
12    }
13    }
14    public static void main(String[] args) {
15        A r=new A();
16    }
17 }
```

5 lines, 84 characters selected

Tab Size: 4 Java



Type here to search



1647
04-02-2021

```
1          /* Constructor Overlaoding */  
2 class A  
3 {  
4     int a; double b; String c;  
5     A()  
6     {  
7         a=100; b=45.32; c="ankit";  
8     }  
9     A(int x)  
10    {  
11    }  
12    }  
13    A(double y, String z)  
14    {  
15    }  
16    }  
17 }
```



```
7         a=100; b=45.32; c="ankit";
8     }
9     A(int x)
10    {
11    }
12    A(double y,String z)
13    {
14    }
15    }
16 }
17 }
18 class B
19 {
20     A r=new A();
21     A r2=new A(10);
22     A r3=new A(23.89,"Ankush");
23 }
```

```
10     {
11         a=x;
12     }
13     A(double y, String z)
14     {
15         b=y; c=z;
16     }
17 }
18 class B
19 {
20     A r=new A();
21     A r2=new A(10);
22     A r3=new A(23.89, "Ankush");
23     System.out.println(r.a+" "+r.b+" "+r.c);
24     System.out.println(r2.a);
25     System.out.println(r3.b+" "+r3.c);
26 }
```

```
1      /* Constructor Overloading */
2  class A
3  {
4      int a; double b; String c;
5      A()
6      {
7          a=100; b=45.32; c="ankit";
8      }
9      A(int x)
10     {
11         a=x;
12     }
13     A(double y, String z)
14     {
15         b=y; c=z;
16     }
17 }
18 class B
19 {
20     public static void main(String[] args) {
21
22         A r=new A();
23         A r2=new A(10);
24         A r3=new A(23.89, "Ankush");
25         System.out.println(r.a+" "+r.b+" "+r.c);
26     }
27 }
```

```
B.java:25: error: <identifier> expected
        System.out.println(r3.b+" "+
```

6 errors

```
C:\Users\WIN10\Desktop>javac B.java
```

C:\Users\WIN10\Desktop>java B

100 45.32 anki

10

23.89 Ankush

C:\Users\WIN10\Desktop>

```
4     int a; double b; String c;
5     private A()
6     {
7         a=1050; b=4.32; c="ahjdt";
8     }
9     A(int x)
10    {
11        a=x;
12    }
13    A(double y,String z)
14    {
15        b=y; c=z;
16    }
17    public static void main(String[] args)
18    {
19        A r=new A();
20        A r2=new A(10);
21        A r3=new A(23.89,"Ankush");
22        System.out.println(r.a+" "+r.b+" "+r.c);
23        System.out.println(r2.a);
24        System.out.println(r3.b+" "+r3.c); C:\Users\WIN10\Desktop>
25    }
26 }
27 }
```

Command Prompt

```
Exception in thread "main" java.lang.IllegalAccessException
: class B tried to access private method 'void A.<init>()
' (B and A are in unnamed module of loader 'app')
at B.main(B.java:22)
```

```
C:\Users\WIN10\Desktop>javac A.java
```

```
C:\Users\WIN10\Desktop>java A
1050 4.32 ahjdt
10
```

```
23.89 Ankush
```



Q. static block? With syntax & Example.

Ans → Static block is such kind of block in java which gets executed at the time of loading the .class file into JVM memory.

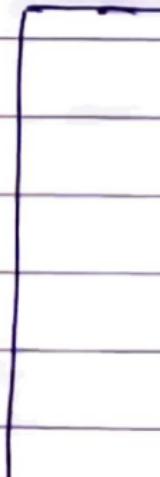
A.java



javac



A.class



class loader

Byt

E

SUBSCRIBE

Reading the .class file into memory

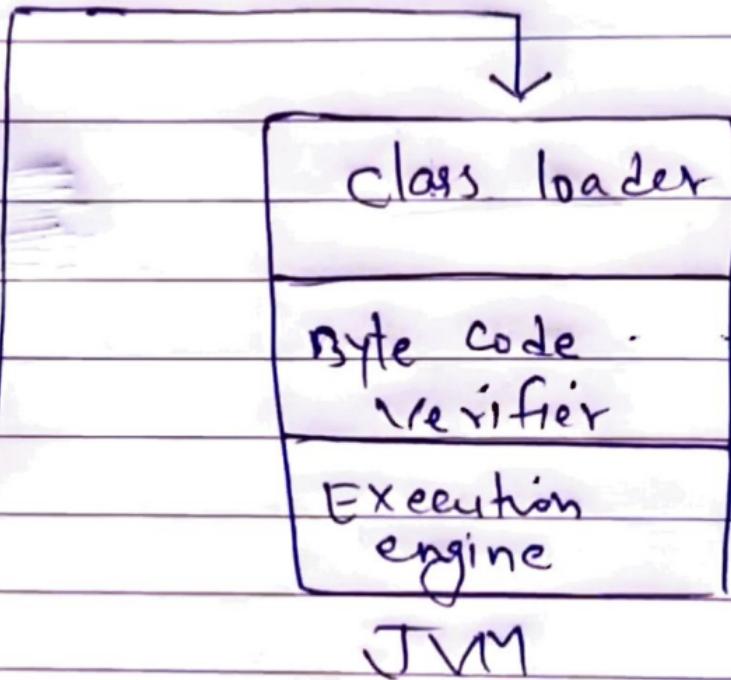
A.java



javac



A.class



SUBSCRIBE

Syntax :-

class A
{

 Static
 {

}

}

Page No.

SUBSCRIBE

```
A.java
```

```
1          /* Static Block */  
2 class A  
3 {  
4     static{  
5         System.out.println("Learn Coding");  
6     }  
7     public static void main(String[] args){  
8         System.out.println("Hello World");  
9     }  
10 }
```

Command Prompt

```
Microsoft Windows [Version 10.0]  
(c) 2019 Microsoft Corporation.  
C:\Users\WIN10>cd desktop  
C:\Users\WIN10\Desktop>javac A.java  
C:\Users\WIN10\Desktop>java A  
Learn Coding  
Hello World  
C:\Users\WIN10\Desktop>
```

```
A.java
```

```
/* Static Block */  
1 class A  
2 {  
3     static{  
4         System.out.print]  
5             Learn Coding  
6     }  
7 }  
8 }
```

```
Select Command Prompt  
C:\Users\WIN10\Desktop>java A  
Learn Coding  
  
C:\Users\WIN10\Desktop>javac A.java  
  
C:\Users\WIN10\Desktop>java A  
Error: Main method not found in class A, please def:  
the main method as:  
    public static void main(String[] args)  
or a JavaFX application class must extend javafx.appli  
cation.Application  
  
C:\Users\WIN10\Desktop>
```



A screenshot of a Sublime Text editor window titled "Ajava". The code in the editor is as follows:

```
4
5     public static void main(String[] args) {
6         A r=new A();
7     }
8     A()
9     {
10    System.out.println("please share");
11 }
12 {
13    System.out.println("please Like");
14 }
15 static{
16    System.out.println("Learn Coding");
17 }
```

To the right of the editor, a terminal window titled "Select Command Prompt" shows the output of running the Java program:

```
C:\Users\WIN10\Desktop>java A
Learn Coding
please Like
please share
```

The terminal window has a yellow circle highlighting the command prompt line.

Line 10, Column 41

 Type here to search



17:50 06-02-2021

```
4
5     public static void main(String[] args) {
6         A r=new A();
7     }
8     A()
9     {
10        System.out.println("please share");
11    }
12    {
13        System.out.println("please Like");
14    }
15    static{
16        System.out.println("Learn Coding");
17    }
18 }
```

Select Command Prompt

```
C:\Users\WIN10\Desktop>java A
Learn Coding
please Like
please share
```

```
C:\Users\WIN10\Desktop>
```

Date

V N / I Instance block

Saathi

Q. What is instance block? With example.

Ans → Instance block is similar to method which has no name, it can be written inside a class only but not inside a method.

Note:- → It always executes before the constructor.

Note:- i) It always executed before the constructor.

ii) We can use variable only inside the instance block not method.

iii) Write time consuming code inside a instance block like - JDBC Connectivity.

Syntax:-

Connectivity.

Syntax:-

class A
{



 // Code

```
1 // INSTANCE BLOCK
2 class A
3 {
4     int a,b;
5     A()
6     {
7         a=30; b=40;
8         System.out.println(a+" "+b);
9     }
10    {
11        a=10;  b=20;
12        System.out.println(a+" "+b);
13    }
14 }
15 }
16 class
```

```
1      /* Instance Block */  
2 class A  
3 {  
4     int a,b;  
5     A()  
6     {  
7         a=30; b=40;  
8         System.out.println(a+" "+b);  
9     }  
10    {  
11        a=10; b=20;  
12        System.out.println(a+" "+b);  
13    }  
14 }  
15 }  
16 class B  
17 {
```

Select Command Prompt

C:\Users\WIN10\Desktop>javac B.java

C:\Users\WIN10\Desktop>java B

10 20
30 40

C:\Users\WIN10\Desktop>



```
B.java
4 int a,b;
5 void show()
6 {
7     a=50; b=60;
8     System.out.println(a+" "+b);
9 }
10 A()
11 {
12     a=30; b=40;
13     System.out.println(a+" "+b);
14 }
15 {
16     a=10; b=20;
17     System.out.println(a+" "+b);
18 }
19 }
20 }
21 class B
22 {
23     public static void main(String[] args) {
24
25         A r=new A();
26         r.show();
```

Line 26, Column 18

Tab Size: 4 Java



Type here to search



08:11 06-02-2021

```
4 int a,b;
5 void show()
6 {
7     a=50; b=60;
8     System.out.println(a+" "+b);
9 }
10 A()
11 {
12     a=30; b=40;
13     System.out.println(a+" "+b);
14 }
15
16 {
17     a=10; b=20;
18     System.out.println(a+" "+b);
19 }
20 }
21 class B
22 {
23     public static void main(String[] args) {
24
25         A r=new A();
26         r.show();
```

Select Command Prompt

C:\Users\WIN10\Desktop>java B
10 20

30 40

C:\Users\WIN10\Desktop>javac B.

C:\Users\WIN10\Desktop>java B
10 20

30 40

50 60

C:\Users\WIN10\Desktop>



```
A.java
```

```
1             /* Instance Vs Static Blocks */  
2 class A  
3 {  
4     //instance  
5     System.out.println("LearnCoding");  
6 }  
7 public static void main(String[] args)  
8 {  
9     A r=new A();  
10 }
```

Select Command Prompt

```
C:\Users\WIN10\Desktop>javac A.java  
C:\Users\WIN10\Desktop>java A  
C:\Users\WIN10\Desktop>javac A.java  
C:\Users\WIN10\Desktop>java A  
LearnCoding  
C:\Users\WIN10\Desktop>
```



```
1  /* Instance Vs Static Blocks */  
2  class A  
3  {  
4      A()  
5      {  
6          System.out.println("default constructor");  
7      }  
8      //instance  
9      System.out.println("LearnCoding");  
10 }  
11 public static void main(String[] args)  
12     A r=new A();  
13 }  
14 }
```

Select Command Prompt

C:\Users\WIN10\Desktop>javac A.

C:\Users\WIN10\Desktop>java A
LearnCoding

C:\Users\WIN10\Desktop>javac A.

C:\Users\WIN10\Desktop>java A
LearnCoding
default constructor

C:\Users\WIN10\Desktop>

V.V.I

Instance Vs Static Block

Q. Difference between Instance & Static block?

Ans→

Instance

① It deals with object.

② Executed at the time of object creation.

With Program

③ No any keyword required.

④ Static & non-static variable can be accessed

Static

① It deals with class.

② Executed at the time of loaded .class file in JVM

With Program

③ Static keyword is required.

④ Only static variable can be accessed



Type here to search

