

#Static

Any method or variable can be static.

↳ Accessible through same class or other classes without creating an object

↳ 'classname.member'

Static method/ variable

{ works like a global variable }

Class load इलेस मेमर्य

20A नम्बर = do A

Allocate डॉफ्ट

"o, do" do

Static methods and variables can be shared just like global variables.

class A {

 static int a = 10;

 static int b = 20;

 int c = 30; ←

 static void show() {

 a = 20 तक होते हैं

We can't directly access

this variable since it is not static

 System.out.println("a: " + a + "\n" + b + ");

}

 static {

(ब्रॉडसिलिंग प्रोसेस)

 System.out.println("Hello.");

}

on class loading,

load static values, execute static blocks.

public class Main

public static void main(String[] args){

A.show();

}

to access the int variable "c" we
need to create an object of A.

A ob = new A();

⇒ ob.c

< static variables are assignable. >

can be changed at runtime

values

final int x = 20;

cannot assign a new value.

can be assigned only once
(during initialization)

< Static has no this pointer >

< Static do not support "super" keyword >

< Static methods can be overloaded >

~~#~~ Inheritance ~~#~~

एक class एक inherit करता है, जो

{ Super Class,
Base Class,
Parent class.

दूसरे class inherit करते हैं जो

{ Sub class
derived class
Child class

<private attributes cannot be inherited>

<'default' acts like 'public' >

different package में under ->

Inheritance using Methods

class A{

int i,j; { i("A") having two methods } A

void show1(){

System.out.println("i: "+i+"\n j: "+j);

}

}

{ i("A") having two methods } A

public class Main

public static void main
(String[] args){

B ob = new B();

B.set(10,20,30);

B.show();

B.show1();

class B extends A{

int k;

void set(int i,int j,int k){

this.i=i;

this.j=j;

this.k=k;

}

void show(){

System.out.println(i+j+k);

Default constructor
calling sequence:

Super class

↳ Then subclass

A::

↓
B::

}

{ obj1.d0 = obj1? (d0,0) and A

{ obj1.d0 = obj1

obj1.d0 = obj1.d0 + obj2.d0 + obj3.d0 + ... + objn.d0

Class A

int i,j;

```
A } { System.out.println("A"); }
```

13

class B extends A{

```
BU{ System.out.println("B"); }
```

and the following

3600 [3 points])

\Rightarrow Terminal:

A

$(0.2, 0.5, 0.7)$ **B**

words.g

#inheritance

using constructor

Class A Area

~~int height, width;~~

Area { $h=2, w=2$ }

```
Area(int height, int width) { this.height = height;  
    this.width = width; }
```

Area (Area Ob) { height = Ob. height; width = Ob. width; }

```
void show() { System.out.println(height + width); }
```

class Rectangle extends Area {

 Rectangle() {

 super();

}

 Rectangle(int height, int width) {

 super(height, width);

}

 Rectangle(Rectangle ob) {

 super(ob);

}

; height = height, width

; width = width, height

; height = height, width

public class Main {

 public static void main(String[] args) {

 Rectangle ob1 = new Rectangle();

 ob1.show();

 Rectangle ob2 = new Rectangle(4, 2);

 ob2.show();

 Rectangle ob3 = new Rectangle(ob2);

 ob3.show();

}

I question will be exceptional,
max will be scenario-based
questions

Superclass: Box → private height, width, depth
Subclass: BoxL → method volume()
private weight

Class Box {

private double height, width, depth;

Box() {

height = width = depth = 3;

}

Box(double height, double width, double depth) {

this.height = height;

this.width = width;

this.depth = depth;

}

Box(Box ob) {

height = ob.height;

width = ob.width;

depth = ob.depth;

}

void volume() {

System.out.println(

"Volume: " + (height * width * depth));

}

}

```

class Box1 extends Box {
    private double weight;
    double weight weight;
    Box1() { Super(); } this.weight = weight; double weight
    Box1( double height, double width, double depth)
        { Super( height, width, depth); } this.weight=weight
    Box1(Box ob) {
        Super( ob);
    }
    void show() { System.out.println( weight); }
}

```

```

public class Main {
    public static void main(String[] args) {
        Box1 ob1 = new Box1();
        ob1.volume(); System.out.println("Volume is " + volume());
        20
        Box1 ob2 = new Box1(5, 10, 15);
        ob2.volume();
        Box ob3 = new Box1( ob2);
        ob3.volume(); ob3.show();
    }
}

```