

JAVA Assignment

XANDITÀ®

JAVA Assignment



Exercise BOOK

Class Employee

```
{ public static void main (String [] args )  
{ int Employee_id = 14400119039;  
    double sal = 50000.00;  
    String Grade = " 1st grade";  
    System.out.println (" Employee ID is " + Employee_id );  
    System.out.println (" Salary of the Employee : " + sal );  
    System.out.println (" Salary grade : " + Grade );  
}
```

Output:-

Employee ID is 14400119039

Salary of the Employee : 50000.0

Salary grade: 1st Grade

class Cloth

```
{ public static void main (String [] args )  
{ String colour = " pink ";  
    double cost = 1200.00;  
    String size = " XL ";  
    System.out.println (" The colour of the cloth is " + colour );  
    System.out.println (" The cost of the cloth is " + cost );  
    System.out.println (" The size of the cloth is : " + size );  
}
```

Output:-

The colour of the cloth is pink

The cost of the cloth is 1200.0

The size of the cloth is : XL

For loop

1) class Sample-2

```

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

    for (int i=55; i<=65; i++) {
        system.out.println (i);
    }
}

```

10times

output

55
56
57
58
59
60
61
62
63
64
65

i = 55	i <= 65	i++
55	55 <= 65	55+1 = 56
56	56 <= 65	56+1 = 57
57	57 <= 65	57+1 = 58
58	58 <= 65	58+1 = 59
59	59 <= 65	59+1 = 60
60	60 <= 65	60+1 = 61
61	61 <= 65	61+1 = 62
62	62 <= 65	62+1 = 63
63	63 <= 65	63+1 = 64
64	64 <= 65	64+1 = 65
65	65 <= 65	65+1 = 66
66	66 <= 65 F	X

2. write a program to display 89 to 79.

class Sample-3

```

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

    for (int i = 89; i >= 79; i--) {
        system.out.println (i);
    }
}

```

10times

output

89
88
87
86
85
84
83
82
81
80
79

i = 89	i >= 79	i --
89	89 >= 79 T	89 - 1 = 88
88	88 >= 79 T	88 - 1 = 87
87	87 >= 79 T	87 - 1 = 86
86	86 >= 79 T	86 - 1 = 85
85	85 >= 79 T	85 - 1 = 84
84	84 >= 79 T	84 - 1 = 83
83	83 >= 79 T	83 - 1 = 82
82	82 >= 79 T	82 - 1 = 81
81	81 >= 79 T	81 - 1 = 80
80	80 >= 79 T	80 - 1 = 79
79	79 >= 79 T	79 - 1 = 78
78	78 >= 79 F	X

Q) write two program to print 55-65 odd, EVEN. using if

Source code

class Sample-odd

```
javm { public static void main (String [] args) {  
    for (int i = 55; i <= 65; i++) {  
        if (i % 2 != 0) {  
            System.out.println (i);  
        }  
    }  
}
```

output:-

Output:-

55
57
59
61
63
65

Tracing

i = 55	i ≤ 65	i++	i % 2 != 0
55	55 ≤ 65 T	55 + 1 = 56	55 % 2 != 0, T
56	56 ≤ 65 T	56 + 1 = 57	56 % 2 != 0 F
57	57 ≤ 65 T	57 + 1 = 58	57 % 2 != 0, T
58	58 ≤ 65 T	58 + 1 = 59	58 % 2 != 0 F
59	59 ≤ 65 T	59 + 1 = 60	59 % 2 != 0 T
60	60 ≤ 65 T	60 + 1 = 61	60 % 2 != 0 F
61	61 ≤ 65 T	61 + 1 = 62	61 % 2 != 0, T
62	62 ≤ 65 T	62 + 1 = 63	62 % 2 != 0, F
63	63 ≤ 65 T	63 + 1 = 64	63 % 2 != 0, T
64	64 ≤ 65 T	64 + 1 = 65	64 % 2 != 0, F
65	65 ≤ 65 T	65 + 1 = 66	65 % 2 != 0, T
66	66 ≤ 65 F	X	X

Source code

class Sample-EVEN

```
javm { public static void main (String [] args) {  
    for (int i = 55; i ≤ 65; i++) {  
        if (i % 2 == 0) {  
            System.out.println (i);  
        }  
    }  
}
```

Output

56
58
60
62
64

Tracing

$i = 55$	$i \leq 65$	$i++$	$i \% 2 == 0$
55	$55 \leq 65$ T	$55 + 1 = 56$	$55 \% 2 == 0, F$
56	$56 \leq 65$ T	$56 + 1 = 57$	$56 \% 2 == 0, T$
57	$57 \leq 65$ T	$57 + 1 = 58$	$57 \% 2 == 0, F$
58	$58 \leq 65$ T	$58 + 1 = 59$	$58 \% 2 == 0, T$
59	$59 \leq 65$ T	$59 + 1 = 60$	$59 \% 2 == 0, F$
60	$60 \leq 65$ T	$60 + 1 = 61$	$60 \% 2 == 0, T$
61	$61 \leq 65$ T	$61 + 1 = 62$	$61 \% 2 == 0, F$
62	$62 \leq 65$ T	$62 + 1 = 63$	$62 \% 2 == 0, T$
63	$63 \leq 65$ T	$63 + 1 = 64$	$63 \% 2 == 0, F$
64	$64 \leq 65$ T	$64 + 1 = 65$	$64 \% 2 == 0, T$
65	$65 \leq 65$ T	$65 + 1 = 66$	$65 \% 2 == 0, F$
66	$66 \leq 65$ F	X	X

write two program to print 75-65 odd & even no.

Source code

class SampleOdd

```
{
    public static void main (String [] args)
    {
        for (int i = 75; i >= 65; i--)
        {
            if (i % 2 != 0)
                System.out.println(i);
        }
    }
}
```

Output

```

75
73
71
69
67
65

```

Tracing

$i = 75$	$i >= 65$	$i--$	$i \% 2 != 0$
75	$75 >= 65$ T	$75 - 1 = 74$	$75 \% 2 != 0, T$
74	$74 >= 65$ T	$74 - 1 = 73$	$74 \% 2 != 0, F$
73	$73 >= 65$ T	$73 - 1 = 72$	$73 \% 2 != 0, T$
72	$72 >= 65$ T	$72 - 1 = 71$	$72 \% 2 != 0, F$
71	$71 >= 65$ T	$71 - 1 = 70$	$71 \% 2 != 0, T$
70	$70 >= 65$ T	$70 - 1 = 69$	$70 \% 2 != 0, F$
69	$69 >= 65$ T	$69 - 1 = 68$	$69 \% 2 != 0, T$
68	$68 >= 65$ T	$68 - 1 = 67$	$68 \% 2 != 0, F$
67	$67 >= 65$ T	$67 - 1 = 66$	$67 \% 2 != 0, T$
66	$66 >= 65$ T	$66 - 1 = 65$	$66 \% 2 != 0, F$
65	$65 >= 65$ T	$65 - 1 = 64$	$65 \% 2 != 0, T$
64	$64 >= 65$ F	X	X

Select class3 Even

```
public static void main( String[ ] args )  
{  
    for( int i = 75; i >= 65; i-- )  
    {  
        if ( i % 2 == 0 )  
        {  
            System.out.println( i );  
        }  
    }  
}
```

Output
74
72
70
68
66

Tracing

$i = 75$	$i >= 65$	$i--$	$i \% 2 = 0$
75	$75 >= 65 \text{ T}$	$75 - 1 = 74$	$75 \% 2 = 0 \text{ F}$
74	$74 >= 65 \text{ T}$	$74 - 1 = 73$	$74 \% 2 = 0, \text{ T}$
73	$73 >= 65 \text{ T}$	$73 - 1 = 72$	$73 \% 2 = 0, \text{ F}$
72	$72 >= 65 \text{ T}$	$72 - 1 = 71$	$72 \% 2 = 0, \text{ T}$
71	$71 >= 65 \text{ T}$	$71 - 1 = 70$	$71 \% 2 = 0, \text{ F}$
70	$70 >= 65 \text{ T}$	$70 - 1 = 69$	$70 \% 2 = 0, \text{ T}$
69	$69 >= 65 \text{ T}$	$69 - 1 = 68$	$69 \% 2 = 0, \text{ F}$
68	$68 >= 65 \text{ T}$	$68 - 1 = 67$	$68 \% 2 = 0, \text{ T}$
67	$67 >= 65 \text{ T}$	$67 - 1 = 66$	$67 \% 2 = 0, \text{ F}$
66	$66 >= 65 \text{ T}$	$66 - 1 = 65$	$66 \% 2 = 0, \text{ T}$
65	$65 >= 65 \text{ T}$	$65 - 1 = 64$	$65 \% 2 = 0, \text{ F}$
64	$64 >= 64 \text{ F}$	X	X

1
0
1

1 0 0

Method

① WAP to print area of a triangle.

class Triangle

{

 static void area()

{

 int b = 10;

 int h = 12;

 int A = b * h / 2;

 System.out.println(A);

}

 public static void main(String[] args)

{

 System.out.println(" *** AREA of the Triangle");

 area();

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

}

output:- *** AREA of the Triangle

 60

 *** ENDS ***

② WAP to print area of of a Trapezoid

class Trapezoid

{

 static void area()

{

 int a = 8;

 int b = 6;

 int h = 4;

 int A = (a+b)/2 * h;

 System.out.println(A);

 public static void main(String[] args)

{

 System.out.println(" *** Area of. Trapezoid ***");

 area();

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

}

output:-

 *** Area of Trapezoid

 28

 *** ENDS ***

3) write a program to display area of the square.

class square

```
{  
    static void area()
```

```
{  
    int a = 10;
```

```
    int Area = a*a;
```

```
    System.out.println(Area);
```

```
}
```

```
public static void main (String [] args)
```

```
{  
    System.out.println("**** Area of Square ****");
```

```
} area();  
System.out.println("**** ENDS ****");
```

Output:-

~~System.out.println~~
**** Area of Square ****
100
**** ENDS ****



write a program to display area of parallelogram.

class parallelogram

```
{  
    static void area()
```

```
{  
    int b = 10;
```

```
    int h = 4;
```

```
    int Area = b*h;
```

```
} System.out.println("Area");
```

```
}
```

```
public static void main (String [] args)
```

```
{  
    System.out.println("Area of the parallelogram");
```

```
} area();  
System.out.println("**** ENDS ****");
```

Output:-

Area of the parallelogram
40
**** ENDS ****

5) write a program to display area of a rectangle

```

class Rectangle
{
    static void area()
    {
        int h = 10;
        int w = 6;
        int A = w * h;
        System.out.println(A);
    }

    public static void main (String [] args)
    {
        System.out.println ("* * * Area of the rectangle * * *");
        area();
        System.out.println ("* * * END * * *");
    }
}

```

output:-

Area of the rectangle
60
*** END ***

6) write a program to print Area of a Ellipse.

```

class Ellipse
{
    static void area()
    {
        final double pi = 3.14;
        int a = 6;
        int b = 10;
        double Area = pi * a * b;
        System.out.println (Area);
    }

    public static void main (String [] args)
    {
        System.out.println ("AREA OF THE ELLIPSE");
        area();
    }
}

```

output:-

AREA OF THE ELLIPSE
188.4

Q) write a program to print area of a sector.

```
class Sector
{
    static void area():
    {
        int r = 5;
        int t = 30;
        int Area = r * r / 2 * t;
        System.out.println(Area);
    }
    public static void main(Strings [I args])
    {
        System.out.println("Area of Sector");
    }
}
```

Output:-

** Area of Sector **
360°

class Triangle_P

```
{
    static void area (int b, int h)
    {
        int A = b * h / 2;
        System.out.println (A);
    }
}
```

```
>VM >public static void main (String [] args)
{
    System.out.println ("*** Area of the Triangle ***");
    area (10, 12);
    System.out.println ("*** ENDS ***");
}
```

output:- *** Area of the Triangle
 60
 *** ENDS ***

WAP to display area of a Trapezoid using method with parameter

class Trapezoid_P

```
{
    static void area (int a, int b, int h)
    {
        int A = (a+b) / 2 * h;
        System.out.println (A);
    }
}
```

```
>VM >public static void main (String [] args)
{
    System.out.println ("Area of Trapezoid");
    area (8, 6, 9);
    System.out.println ("*** ENDS ***");
}
```

output:- Area of Trapezoid
 28
 *** ENDS ***

class factorial

```
public static void main (String args[])
{
    int fact = 1;
    for (int i = 7; i > 1; i--)
    {
        fact = fact * i;
    }
}
```

Output:-
5040

Logic :-
7x6x5x4x3x2x1
= 5040

```
System.out.println(fact);
```

Output:-

i = 7	i > 1	i - 1	fact	fact * i
7	7 > 1	7 - 1 = 6	fact = 1	fact = 7x1 = 7
6	6 > 1	6 - 1 = 5	fact = 7	fact = 7x5 = 35
5	5 > 1	5 - 1 = 4	fact = 35	fact = 35x4 = 140
4	4 > 1	4 - 1 = 3	fact = 140	fact = 140x3 = 420
3	3 > 1	3 - 1 = 2	fact = 420	fact = 420x2 = 840
2	2 > 1	2 - 1 = 1	fact = 840	fact = 840x1 = 840
1	1 > 1	1 - 1 = 0	fact = 840	fact = 840x0 = 0
0	0 > 1	x		

class factorial

```
public static void main (String args[])
{
    int fact = 1;
```

```
for (int i = 9; i > 1; i--)
{
    fact = fact * i;
}
```

Output:-

Logic :-
9x8x7x6x5x4x3x2x1
= 362880

```
System.out.println(fact);
```

Output:-

i = 9	i > 1	i - 1	fact	fact * i
9	9 > 1	9 - 1 = 8	fact = 1	fact = 9x1 = 9
8	8 > 1	8 - 1 = 7	fact = 9	fact = 9x7 = 63
7	7 > 1	7 - 1 = 6	fact = 63	fact = 63x6 = 378
6	6 > 1	6 - 1 = 5	fact = 378	fact = 378x5 = 1890
5	5 > 1	5 - 1 = 4	fact = 1890	fact = 1890x4 = 7560
4	4 > 1	4 - 1 = 3	fact = 7560	fact = 7560x3 = 22680
3	3 > 1	3 - 1 = 2	fact = 22680	fact = 22680x2 = 45360
2	2 > 1	2 - 1 = 1	fact = 45360	fact = 45360x1 = 45360
1	1 > 1	1 - 1 = 0	fact = 45360	fact = 45360x0 = 0
0	0 > 1	x		

Output:-
362880

Method with return type

class Triangle

class Square

$$\frac{1}{2} \times b \times h$$

{ static int area()

{ static int area;

int b = 10;

int result;

int a = 8;

int result;

int area = b * a / 2;

int result;

return area;

int result;

return area;

int result;

return result;

Main END

class Rectangle_2

```
{ void area ( int a, int b ) }
```

```
{ int area = a * b ; }
```

```
 S.o.p (area) ;
```

```
} public static void main( string[] )
```

```
{ S.o.p ( " ** Main Start ** " ) .
```

```
 S.o.p ( " ** Main Start ** " ) .
```

```
 new Rectangle_2 ( ). area ( 10, 8 ) .
```

```
 S.o.p ( " ** Main END ** " ) .
```

```
 S.o.p ( " ** Main END ** " ) .
```

```
 } }
```

Output:

```
** Main Start **  
80  
** Main END **
```

Square - 2

```
{ void area ( int a ) }
```

```
{ int area = a * a ; }
```

```
 S.o.p (area) ;
```

```
} public static void main ( string[] )
```

```
{ S.o.p ( " ** Main Start ** " ) .
```

```
 S.o.p ( " ** Main Start ** " ) .
```

```
 new Square_2 ( ). area ( 10 ) .
```

```
 S.o.p ( " ** Main END ** " ) .
```

```
 } }
```

Output:

```
** Main Start **  
100  
** Main END **
```

class Triangle_2

```
{ void area ( int b, int h ) }
```

```
{ int area = b * h / 2 ; }
```

```
 S.o.p (area) ;
```

```
} public static void main( string[] )
```

```
{ S.o.p ( " ** Main Start ** " ) .
```

```
 S.o.p ( " ** Main Start ** " ) .
```

```
 new Triangle_2 ( ). area ( 20, 10 ) .
```

```
 S.o.p ( " ** Main END ** " ) .
```

```
 S.o.p ( " ** Main END ** " ) .
```

```
 } }
```

Output:

```
** Main Start **  
100  
** Main END **
```

Ellipse - 2

```
{ void area ( int a, int b ) }
```

```
{ final double pi = 3.14 ; }
```

```
 S.o.p ( " ** Main Start ** " ) .
```

```
 S.o.p ( " ** Main Start ** " ) .
```

```
 new Ellipse_2 ( ). area ( pi * a * b ) .
```

```
 S.o.p (area) ;
```

```
} public static void main( string[] )
```

```
{ S.o.p ( " ** Main Start ** " ) .
```

```
 S.o.p ( " ** Main Start ** " ) .
```

```
 new Ellipse_2 ( ). area ( 10, 20 ) .
```

```
 S.o.p ( " ** Main END ** " ) .
```

```
 } }
```

Output:

```
** Main Start **  
628.0  
** Main END **
```

using this keyword

constructor assignment

Stack

H.M

class Car - 2

{
String car - brand;

String car - color;

double car - price;

Car - 2 (String car - brand, String car - color, double car - price)

MA

SPA

this . car - Brand = car - Brand;
this . car - color = car - color;
this . car - price = car - price;

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

```
car - 2 s1 = new Car - 2 ("Tata", "red", 50000.0)
S.O.P (s1 . car - Brand);
S.O.P (s1 . car - color);
S.O.P (s1 . car - price);
```

class Mobile - 2

Output

Tata

red

50000.0

{
String mobile - colour;

String mobile - name;

double mobile - price;

Mobile - 2 (String mobile - colour, String mobile - name, double mobile - price)

{
this . mobile - colour = mobile - colour;
this . mobile - name = mobile - name;
this . mobile - price = mobile - price;

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

Mobile - 2 s1 = new Mobile - 2 ("Green", "poco", 14000.00);

S.O.P (s1 . mobile - colour);

S.O.P (s1 . mobile - name);

S.O.P (s1 . mobile - price);

Output

Green

poco

14000.0

class Reverse

{ public static void main(

using this

int fact = 1;

for (int i = 5; i > 0; i--

fact = fact * i;

} System.out.println(fact);

}

class Reverse

variables

i = 5
s1 = "SOURAV"
s2 = " ";

i = 4
s1 = "VARA"
s2 = " ";

i = 3
s1 = "VAR"
s2 = " ";

i = 2
s1 = "V"
s2 = " ";

i = 1
s1 = ""
s2 = " ";

i = 0
s1 = ""
s2 = " ";

for (int i = 5; i > 0; i--)

class Mobile

String

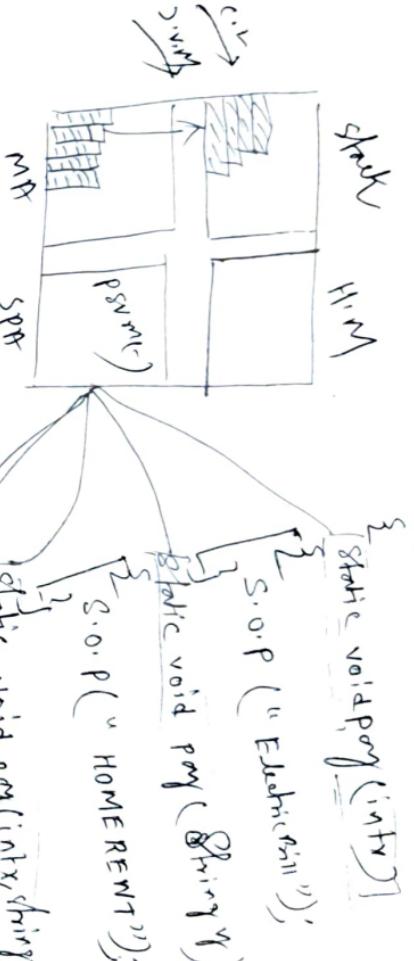
String

double

Mobile

Method overloading program diagram

class paytm



class ~~Google~~

static void search(intx)

static void search(" NEWS")

static void search(Stringy) ~~public~~ static void main(—)

static void search(" Image")

static void search(intx, Stringy)

static void search(" Maps")

static void search(" Website")

static void main(Stringt, intx)

public static void main(Stringt, Stringy)

Output

NEWS

WEBSITE

IMAGE

MAPS

Search(10);
Search("B", 10);
Search("B");
Search(10, "B11");

class Xender

{ void show (int x)

{ show ("Movement");

void show (string y);
void show ("Music");

void show (int x, string y);
void show ("Photo");

} void show (string y, int x)

{ cout << "Music"; }

public static void main (String args[])

{ Xender * x = new Xender();

x->show (20);

x->show ("Hi");

x->show (20, "Hello");

x->show ("Hello", 20);



Method overriding

2nd scenario

class whatsupp-v1

{ void sent()

{ S.O.P("Hey I can send sms"); }

class whatapp-v2 extends whatsupp-v1

{ void sent()

{ S.O.P("Hey I can send money"); }

{ }

class whatapp-updated

{ public static void main(→)

{ whatsupp-v1 w1 = new whatsupp-v1;

{ w1.sent(); }

{ }

class Disney

{ void streaming()

{ S.O.P("Hey I streaming animated video"); }

class Holstar extends Disney

{ S.O.P("Hey void streaming()")

{ super.streaming();

{ S.O.P("Hey I streaming movies, cricket, football "); }

{ }

class Disney-Holstar

{ public static void main(→)

{ Disney-D1 = new Disney(); }

{ D1.streaming(); }

class Face

{ void m.

{ S.O.P(

) T-1

Up - Casting - down casting overwriting

(1) class ~~Airway~~
 {
 void wings()
 {
 S.o.p("Two wings");
 }
 }

class Runway extends Airway
 {
 void wheel()
 {
 S.o.p("Three wheels");
 }
 }

class Aeroplane

{
 public static void main (String args)
 {
 Runway R1 = new Runway();
 Aeroplane A1 = new Aeroplane();
 A1.wings();
 }
}

S.o.p("Up casting **");

~~Runway~~ R1 = (Runway) A1

R1.wings();
R1.wheel();

Output

** Up casting **
two wings

** Down casting **
two wings

Two wings

class Madhyamik

{ void qualification-1 ()

{ S.o.p ("10th pass")'

}

} class Higher_Secondary

{ void qualification-2 ()

{ S.o.p ("12th pass")'

}

} class D.M.

{ D.S.V.M ()

{ S.o.p ("* * up coming **").

~~qualification-1 = new qualification-2~~.
Madhyamik M1 = new Higher - Secondary

M1 . qualification (1)

S.o.p ("* * Down coming **").

Higher_Secondary H1 = (Higher_Secondary) M1

H1 . qualification-1 ()

H1 . qualification-2 ()

}

Output

* up coming *

10th pass

* down coming **

10th pass

12th pass.

Mobile store

Any morphism or assignment

```
void smartphone()
{
    S.o.p("Have many Brand smartphones");
}
```

```
class Samsung
```

```
{ void smartphone()
```

```
{ S.o.p(" price is 25000 ");}
}
```

```
class Vivo
```

```
{ void smartphone()
}
```

```
{ S.o.p(" price is 30000 ");}
}
```

```
class Apple
```

```
{ void smartphone()
}
```

```
{ S.o.p(" price is 150000 ");}
}
```

```
class Shopkeeper
```

```
{ static void Supply(MobileStore s1)
}
```

```
{ S1. smartphone();
}
```

```
class Main
```

```
{ public static void main (String args[])
}

```

```
Samsung n1 = new Samsung();
Vivo v1 = new Vivo();
Apple n1 = new Apple();

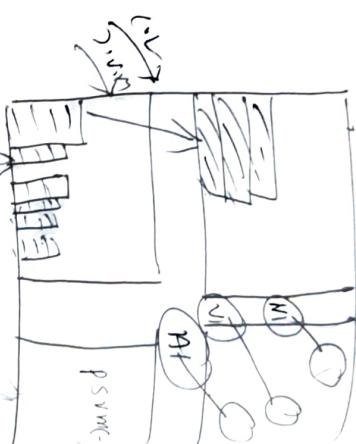
```

```
Shopkeeper. Supply(n1);
Shopkeeper. Supply(v1);
Shopkeeper. Supply(n1);

```

```
Output
price is 25000
price is 30000
price is 150000
price is 25000

```



```
class MobileStore
{
    void S.o.p()
    {
        S.o.p(" price is 25000 ");
        S.o.p(" price is 30000 ");
        S.o.p(" price is 150000 ");
    }
}
```

```
class Samsung
{
    void smartphone()
    {
        S.o.p(" price is 25000 ");
    }
}
```

```
class Vivo
{
    void smartphone()
    {
        S.o.p(" price is 30000 ");
    }
}
```

```
class Apple
{
    void smartphone()
    {
        S.o.p(" price is 150000 ");
    }
}
```

```
class Shopkeeper
{
    static void Supply(MobileStore s1)
    {
        S1. smartphone();
    }
}
```

```
class Main
{
    public static void main (String args[])
    {
        Samsung n1 = new Samsung();
        Vivo v1 = new Vivo();
        Apple n1 = new Apple();
        Shopkeeper. Supply(n1);
        Shopkeeper. Supply(v1);
        Shopkeeper. Supply(n1);
    }
}
```

class Me

 private int count = 0;

 S.o.F (where some behaviors)

} class Home extends Me {

 void break_out()

{ S.o.F ("Innocent") ; }

class School extends Me

 void behaviour()

{ S.o.F ("Student") ; }

class Bank extends Me

 void behaviour()

{ S.o.F ("Passenger") ; }

class Friend

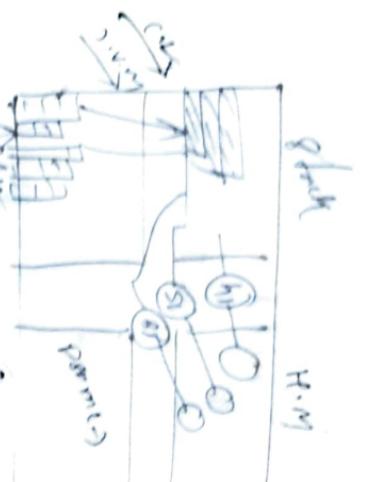
 static void wellKnown(Me f)

{ f . behaviour(); }

class Main

 static void main (String args)

```
    Home h1 = new Home();
    School s1 = new School();
    Bank B1 = new Bank();
    friend wellKnown(h1);
    friend wellKnown(s1);
    friend wellKnown(B1);
```



Absract

interface Mobile-shop

{ void mobile(); }

class Samsung implements mobile.shop
{ public void mobile(); }

class S.O.P(" price is 15000.00 ");

class Apple implements mobile.shop
{ public void mobile(); }

S.O.P(" price is 90000.00 ");

class Vivo implements mobile.shop

class public void mobile();

S.O.P(" price is 25000.00 ");

class ~~Customer~~ shopkeeper

static void sell(Mobile-shop MI)

MI.mobile();

class customer

PSVM(-)

Samsung s1 = new Samsung();

Apple A1 = new Apple();

Vivo V1 = new Vivo();

Shop-keeper. sell(s1);

Shop-keeper. sell(A1);

Shop-keeper. sell(V1);

Output
price is 15000.00
price 15000.00

~~class~~ Interface Doctor

{
 void treatment();

}
class Dentist

{
 public void treatment();

} S.o.p ("Teeth Treatment");

class Cardiologist

{
 public void treatment();

} S.o.p ("Heart Treatment");

class Arthropedic

{
 public void treatment();

} S.o.p ("Bone Treatment");

class Hospital

{
 static void facilities(Doctor D1);

}
 {
 D1.treatment();

}
class Patient

{
 public static void main(String[] args)

}
Dentist D1 = new Dentist();

Cardiologist C1 = new Cardiologist();

Arthropedic A1 = new Arthropedic();

Hospital facilities(D1);

Hospital facilities(C1);

Hospital facilities(A1);

Abstract 2

```
{ interface Editplus
  {
    void template();
  }
}

class C++ implements Editplus
{
  public void template()
  {
    S.O.P ("Template C++ occurred");
  }
}

class Java implements Editplus
{
  public void template()
  {
    S.O.P ("Template for Java occurred.");
  }
}

class HTML implements Editplus
{
  public void template()
  {
    S.O.P ("Template for HTML occurred");
  }
}

class Click
{
  static void select (Editplus s1)
  {
    s1.template();
  }
}

class User
{
  public static void main (String[] args)
  {
    C++ c1 = new C++();
    Java j1 = new Java();
    HTML h1 = new HTML();
    Click.select (c1);
    Click.select (j1);
    Click.select (h1);
  }
}
```

Output

Terminal

Encapsulation

```

class Facebook
{
    private int password(1234);
    public int getpass();
    {
        return pass;
    }
    public void setpass (int password)
    {
        this.password = password;
    }
}

```

```

class user
{
    public static void main (String [ ] args)
    {
        Facebook f1 = new Facebook();
        System.out.println(f1.getpass());
        f1.setpass(5678);
        System.out.println(f1.getpass());
    }
}

```

Example: 2

```

class Email
{
    private String name;
    public String getName()
    {
        return name;
    }
    public void setName (String newname)
    {
        this.name = newname;
    }
}
public class main
{
    public static void main ( )
    {
        Person m1 = new Person();
        m1.setName("John");
        System.out.println(m1.getName());
    }
}

```