

# Basic Fundamental Programming

## Day 4: Sep 2021

Kiran Waghmare



-----  
Day4:24.09.21

Topics: Static methods, Access specifiers & Arrays  
-----

Static method:

-use static keyword with method

//Instance method or normal method

```
int a;  
void display()  
{  
}
```

```
//Abc a1 = new Abc();  
a1.display();
```

//static method

```
static int a;//static variable  
static void display()//static method  
{  
    ++a;  
}
```

```
++a;//CE  
display();
```

CDAC Mumbai : Kiran Waghmare

FileEditSearchViewEncodingLanguageSettingsToolsMacroRunPluginsWindow?

MOM 2021.txtLecture notes Sep21.txtP19.java

MouseSelectTextDrawStampSpotlightEraserFormatUndo

Who can see what you share here? Recording On

1class P19

2{

3int Cube(int x)

4{

5return (x\*x\*x);

6}

7

8static int Cube1(int x)

9{

10int z= x\*x\*x;

11return z;

12}

13

14public static void main(String args[])

15{

16P19 t1 = new P19();

17//Call for Instance method

18System.out.println(t1.Cube(5));

19

20//Call for Static method

21System.out.println(Cube1(6));

22

23}

24}

25

26

27

28

29

30

31

32

stack

Heap

t1

Cube()

Cube1()

Object

static method

CDAC Mumbai : Kiran Waghmare

You are screen sharingStop Share

//Instance method or normal method

```
int a;  
void display()  
{  
}
```

```
//Abc a1 = new Abc();  
a1.display();
```

```
//static method  
static int a;//static variable  
static void display()//static method  
{  
    ++a;  
}
```

```
++a;//CE  
display();
```

Instance method

Instance variable

static method

static variable

Relationship between Instance  
and static members

```
class P20
{
    static int x =100;//static variable
    int y = 200;//instance variable or non-static

    void display()//instance method or normal
    {
        System.out.println("---Non-Staticmethod-----");
        System.out.println(x);
        System.out.println(y);
    }

    static void show()//static method
    {
        System.out.println("---Staticmethod-----");
        System.out.println(x);
        System.out.println(y);//Compilation Error
        //Instance variable cannot be accessed in static method.
    }

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

```
static void show()//static method
{
    System.out.println("---Staticmethod-----");
    System.out.println(x);
    //System.out.println(y);//Compilation Error
    //Instance variable cannot be accessed in static method.
}
```

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

**Access Specifier**

```
P20 t1 = new P20();
System.out.println(x);//static var
System.out.println(t1.y);// non static

t1.display();
show();
```



```
System.out.println("---Staticmethod-----");  
System.out.println(x);  
//System.out.println(y); //Compilation Error  
//Instance variable cannot be accessed in static method.  
}
```

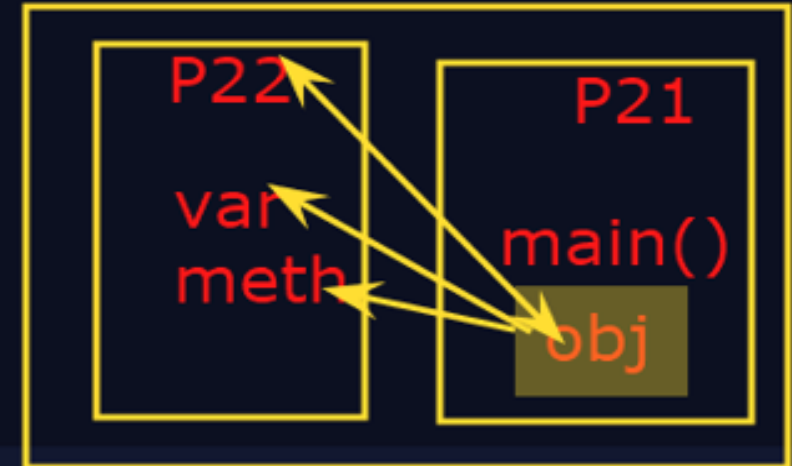
```
}
```

```
class P21{  
    public static void main(String args[])  
    {
```

```
        P21 t1 = new P21();
```

```
        System.out.println(x); //static var  
        System.out.println(t1.y); // non static
```

```
        t1.display();  
        show();
```



```
class Math
{
    static int sum(int x, int y)
    {
        return x+y;
    }

    static void display(int result)
    {
        System.out.println(result);
    }
}
```

Math

var  
fun

P23

main()

```
class P23{
    public static void main(String args[])
    {
        int z=Math.sum(100,200);
        Math.display(z);
    }
}
```



```
public static int sum(int x, int y)
{
    return x+y;
}
```

100+200=300

```
static void display(int result)
{
    System.out.println(result);
}
```

```
class P24{
    public static void main(String args[])
    {
```

```
        int z=Math.sum(100,200);
```

z=300

```
        Math.display(z);
    }
```

CDAC Mumbai : Kiran Waghmare



You are screen sharing



Stop Share

```
class Test
{
```

```
    static void sum(int x, int y)
    {
        int z = x+y;
        System.out.println(z);
    }
```

```
    static void sum(float x, float y)
    {
        int z = x+y;
        System.out.println(z);
    }
```

```
    static void sum(double x, double y)
    {
        int z = x+y;
        System.out.println(z);
    }
```

```
    static void sum(double x, double y, double s)
    {
```

## Method Overloading

### Types

- 1.data types
- 2.no of parameter

## Access Modifiers:

-----

-keywords

-to set the accessibility of variable, methods, constructors, classes, etc.

Types:

### 1.Access Modifiers:

-----

Default:

-No modifiers are required.

Public:

-Visibility to all(world).

Protected:

-Visibility to all packages or subclasses.

Private:

-Visibility to the class.

### 2:Non-Access Modifiers:

-----

static

abstract

synchronized

native

volatile

transient

etc..

# Access Modifiers:

-----  
-keywords

-to set the accessibility of variable, methods, constructors, classes, etc.

Types:

## 1.Access Modifiers:

Default

Public

Protected

Private

## 2:Non-Access Modifiers:

static

abstract

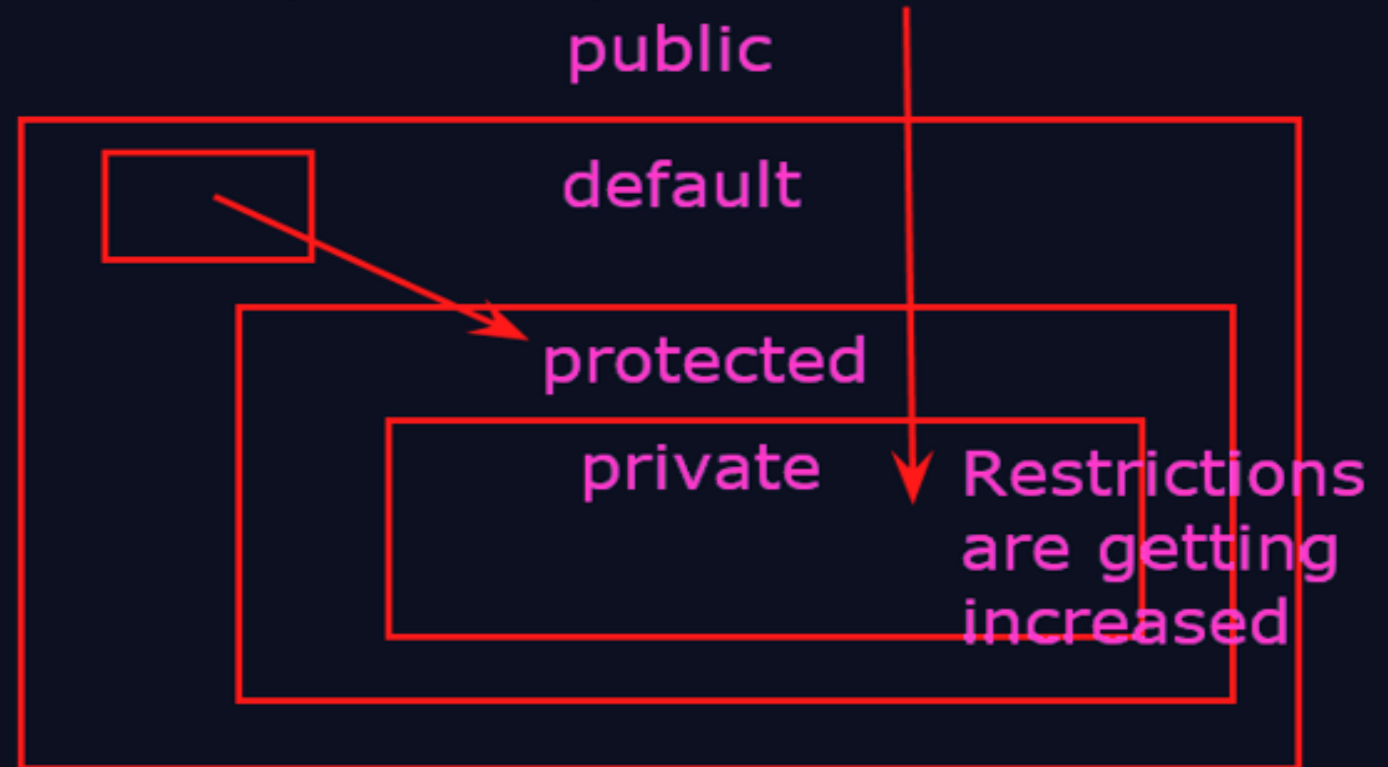
synchronized

native

volatile

transient

etc..



## Array:

-----

-an indexed collection of similar data types.

-homogeneous elements

-fixed size / static data declaration

    e.g: a1[5]: static

        a1[ ]: dynamic

-array is an object.

## Syntax:

### Method 1:

-----

#### 1.Array declaration

-----

<data type> <array name> [];

e.g: int a1[];

#### 2.creation of an array

-----

<array name> = new <data type> [size];

e.g: a1 = new int[5];

or

### Method 2:

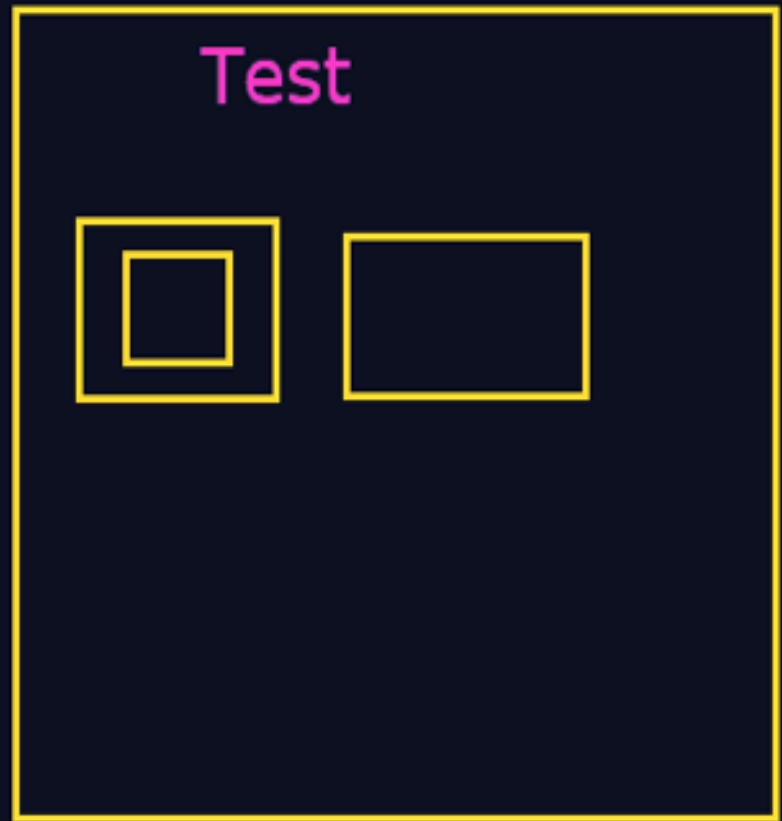
-----

CDAC Mumbai : Kiran Waghmare

<data type> <array name> [] = new <data type> [size];

e.g: int a1[] = new int[5];

```
class Test
{
    protected int x;
}
class P29{
    public static void main(String args[])
    {
        Test t1 = new Test();
        t1.x=40;
        System.out.println(t1.x);
    }
}
```





Binary to decimal:

```
int [] b1 = new int [32];  
int i=0;  
while(n>0)  
{  
    b1[i] = n%2;  
    n= n/2;  
    i++;  
}
```

