



Concepts of Programming

Day 6: Sep 2022

Introduction to Java

Kiran Waghmare

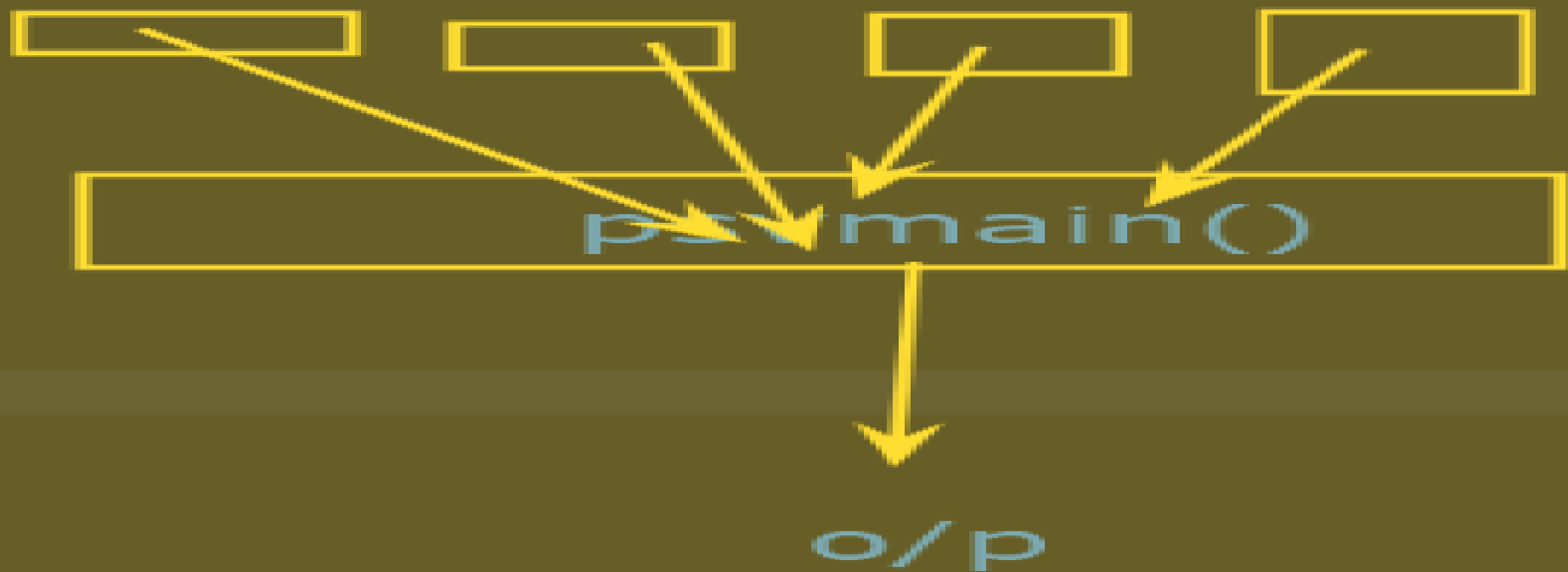
CDAC Mumbai



```
class Operation1{  
    static int sum(int i, int j)  
    {  
        return i+j;  
    }  
}
```



```
class Operation2{  
    public static void main(String args[]){  
  
        System.out.println(Operation1.sum(11,22));  
  
    }  
}
```



userdefined package

class Operation1{

private int x;

public static int sum(int i, int j)

{

return i+j;

}

public static int sum(int i, int j, int k)

{

return i+j+k;

}

}

class Operation3{

public static void main(String args[]){

//Operation1 op1 =new Operation1();

System.out.println(Operation1.sum(11,22));

}

}



Who can see what you share here? Recording On

Method overloading

1. no of parameters
2. data types

class Operation1{

private int x;

public static int sum(int i, int j)

{
return i+j;
}

//No of parameters

public static int sum(int i, int j, int k)

{
return i+j+k;
}

}

class Operation3{

public static void main(String args[]){

//Operation1 op1 =new Operation1();

System.out.println(Operation1.sum(11,22));

}

}

class Operation1{
 private int x;
 public static int sum(int i, int j)
 {
 return i+j;
 }
 //No of parameters
 public static int sum(int i, int j, int k)
 {
 return i*j*k;
 }
}

class Operation3{
 public static void main(String args[]){
 //Operation1 op1 =new Operation1();
 System.out.println(Operation1.sum(11,22,33));
 }
}

Who can see what you share here? Recording On

C:\Test>javac Operation2.java

C:\Test>java Operation2
33

C:\Test>javac Operation3.java

C:\Test>java Operation3
33

C:\Test>javac Operation3.java

C:\Test>java Operation3
7986

C:\Test>

```
class Operation1{
    private int x;
    public static int sum(int i, int j)
    {
        return i+j;
    }
    //No of parameters
    public static int sum(int i, int j, int k)
    {
        return i*j*k;
    }

    public static double sum(double i, double j, double k)
    {
        return i*j*k;
    }
}
```

```
class Operation3{
    public static void main(String args[]){
        //Operation1 op1 =new Operation1();
        System.out.println(Operation1.sum(11,22,33));
    }
}
```

Who can see what you share here? Recording On

C:\Test>javac Operation3.java

C:\Test>java Operation3
33

C:\Test>javac Operation3.java

C:\Test>java Operation3
7986

C:\Test>javac Operation3.java

C:\Test>java Operation3
7986

C:\Test>

```
class Operation1{
    private int x;
    public static int sum(int i, int j)
    {
        return i+j;
    }
    //No of parameters
    public static int sum(int i, int j, int k)
    {
        return i*j*k;
    }

    public static double sum(double i, double j, double k)
    {
        return i*j*k;
    }
}
```

```
class Operation3{
    public static void main(String args[]){
        //Operation1 op1 =new Operation1();

        System.out.println(Operation1.sum(1.1,2.2,3.3));
    }
}
```

Who can see what you share here? Recording On

C:\Test>javac Operation3.java

C:\Test>java Operation3
7986

C:\Test>javac Operation3.java

C:\Test>java Operation3
7986

C:\Test>javac Operation3.java

C:\Test>java Operation3
7.9860000000000001

C:\Test>


```
class Operation1{
```

```
}
```

```
class Operation3{
```

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

```
        System.out.println(Operation1.sum(1.1,2.2,3));
```

```
    }
```

```
}
```

BankTest

psvmain()

Bank

accno
amt

bal()
dep(0
wit()
minbal()

```
String name;  
int marks;
```

```
//Constructor : Default constructor  
Student()  
{  
    id=111; //initialize default value  
    name="Tushar";  
    marks=89;  
}
```

```
//Constructor : Parameterised constructor  
/*Student(int x)
```

```
{  
    id=x;  
}  
*/  
void display()  
{  
    System.out.println("Id= "+id);  
    System.out.println("Name= "+name);  
    System.out.println("Marks= "+marks);  
}
```

```
public static void main(String args[]){  
    Student s1 = new Student();  
    System.out.println("Students Details:");  
    s1.display();  
}
```

Who can see what you share here? Recording On
method Operation1.sum(int,int,int) is r
(argument mismatch; possible lossy co
method Operation1.sum(double,int,double
(argument mismatch; possible lossy co

Note: Some messages have been simplified; r
to get full output
1 error

C:\Test>javac Student.java

C:\Test>java Student
Students Details:
Id= 111
Name= Tushar
Marks= 89

C:\Test>

static String name;
static int marks;

//default constructor

```
Student()  
{  
    id=111;  
    name="Tushar";  
    marks=89;  
}
```

//parameterised constructor

```
Student(int s1, String s2, int s3)  
{  
    id=s1;  
    name=s2;  
    marks=s3;  
}
```

//static method

```
static void getdata(int id1, String name1, int marks1,  
{  
    id=id1;  
    name=name1;  
    marks=marks1;  
}
```

//non-static method

```
void display()  
{  
    System.out.println("Id= "+id);  
    System.out.println("Name= "+name);  
    System.out.println("Marks= "+marks);  
}
```

public static void main(String args[]){

Student s1 = new Student(555, "Saurav", 91);

Who can see what you share here? Recording On

C:\Test>java Student

Students Details:

Id= 111

Name= Tushar

Marks= 89

C:\Test>javac Student.java

C:\Test>java Student

Id= 555

Name= Saurav

Marks= 91

Students Details:

C:\Test>

```
id=111;
name="Tushar";
marks=89;
}

//parameterised constructor
Student(int s1, String s2, int s3)
{
    id=s1;
    name=s2;
    marks=s3;
}

//static method
static void getdata(int id1, String name1, int marks1)
{
    id=id1;
    name=name1;
    marks=marks1;
}

//non-static method
void display()
{
    System.out.println("Id= "+id);
    System.out.println("Name= "+name);
    System.out.println("Marks= "+marks);
}

public static void main(String args[]){
    Student s1 = new Student();
    s1.display();
    Student s2 = new Student(555,"Saurav", 91);
    s1.display();

    System.out.println("Students Details:");
    //getdata(333,"Pragati",90);
    //getdata(111,"Tushar",89);
}
```

Who can see what you share here? Recording On

Marks= 91
Students Details:

C:\Test>javac Student.java

C:\Test>java Student

Id= 111
Name= Tushar
Marks= 89

Id= 555
Name= Saurav
Marks= 91
Students Details:

C:\Test>

Constructor overloading


```
1 class Student{
2     int id;
3     String name;
4     int marks;
5
6     //default constructor
7     Student()
8     {
9         id=111;
10        name="Tushar";
11        marks=89;
12    }
13    //parameterised constructor
14    Student(int id, String name, int marks)
15    {
16        id=id;
17        name=name;
18        marks=marks;
19    }
20    //static method
21
22    public static void main(String args[]){
23
24        Student s1 = new Student();
25        s1.display();
26        Student s2 = new Student(555, "Saurav", 91);
27        s2.display();
28
29        Student s3 = new Student(333, "Pragati", 90);
30        s2.display();
31
32        System.out.println("Students Details:");
33        //getdata(333, "Pragati", 90);
34        //getdata(555, "Saurav", 91);
35    }
```

Instance variable

parameters

```
Student1(int id, String name)
```

```
{  
    this.id=id;  
    this.name=name;  
}  
//static method  
void display()  
{  
    System.out.println("Id= "+id);  
    System.out.println("Name= "+name);  
    System.out.println("Marks= "+marks);  
}
```

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

```
    Student1 s1 = new Student1();  
    s1.display();  
    Student1 s2 = new Student1(555,"Saurav");  
    s2.display();
```

```
    Student1 s3 = new Student1(333,"Pragati");  
    s3.display();
```

```
    System.out.println("Students Details:");  
    //getdata(333,"Pragati",90);  
    //s1.display();
```

```
}
```

Who can see what you share here? Recording On

Command Prompt

1 error

C:\Test>javac Student1.java

C:\Test>java Student1

Id= 111
Name= Tushar
Marks= 89

Id= 555
Name= Saurav
Marks= 91

Id= 333
Name= Pragati
Marks= 0

Students Details:

C:\Test>

-Array is an object.

-Define an array:

-1.Declare array variable

-2.Create an array

Syntax:

-1.Declare array variable

```
<data type> <arrayname>[];
```

or

```
<data type> []<arrayname>;
```

-2.Create an array

```
<arrayname> = new <data type> [size];
```

-Array declaration and creation can be combined:

```
<data type> <arrayname>[] = new <data type>[size];
```

Example:

★
`int a1[] = new int[5];`

fixed size

a1

values:	10	20	30	40	50	
index:	0	1	2	3	4	

int : 4 bytes

size: 5

ml= $4 * 5 = 20$ bytes

`a1[0] = 10;`

★ Example:

`int a1[] = new int[5];` fixed size

`a1`

values:

10

20

30

40

50

`int : 4 bytes`

`size: 5`

`ml = 4 * 5 = 20 bytes`

index:

0

1

2

3

4

`a1[0] = 10;`

`ArrayIndexOutOfBoundsException`

`int a1 = {2,4,6,8,9,10};`

array initializer

```
int a1[] = new int[20];  
System.out.println(a1.length);
```

```
class Array1{
```

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

```
int[] a1;  
a1=new int[5];
```

```
a1[0]=10;  
a1[1]=20;  
a1[2]=30;  
a1[3]=40;  
a1[4]=50;
```

```
for(int i=0;i<=4  
{  
    System.out.println("Index "+i+"="+a1[i]);  
}
```

```
}
```

```
}
```

Command Prompt

C:\Test>javac Array1.java

C:\Test>java Array1

Index 0=10 ✓

Index 1=20 ✓

Index 2=30 ✓

Index 3=40 ✓

Index 4=50 ✓

C:\Test>

a1

0

10 ✓

1

20 ✓

2

30 ✓

3

40 ✓

4

50 ✓

5

0
1
2
3
4
5

0
1
2
3
4

```
class Array1{
```

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

```
int[] a1;  
a1=new int[5];
```

```
a1[0]=10;  
a1[1]=20;  
a1[2]=30;  
a1[3]=40;  
a1[4]=50;
```

```
for(int i=0;i<=4  
{  
    System.out.println("Index "+i+"="+a1[i]);  
}
```

```
}
```

```
}
```

Command Prompt

C:\Test>javac Array1.java

C:\Test>java Array1

Index 0=10 ✓

Index 1=20 ✓

Index 2=30 ✓

Index 3=40 ✓

Index 4=50 ✓

C:\Test>

a1

0

10 ✓

1

20 ✓

2

30 ✓

3

40 ✓

4

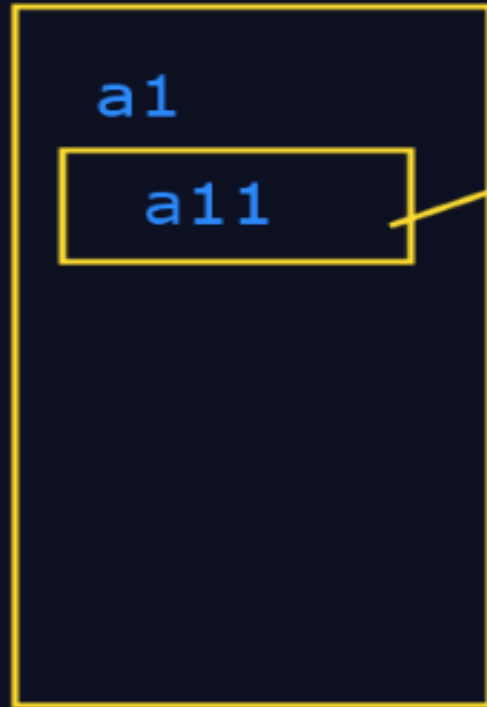
50 ✓

5

0
1
2
3
4
5

0
1
2
3
4

```
int a1[] = new int[5];
```



main method stack

A diagram of an array, labeled 'a11' at the top right. The array is represented as a vertical column of five boxes, each containing a value. The indices 0 through 5 are listed on the left side of the boxes. The values are 10, 20, 30, 40, and 50. The first three values (10, 20, 30) have a checkmark next to them. An arrow points from the 'a11' box in the main method stack to the first element of the array (index 0).

0	10 ✓
1	20 ✓
2	30 ✓
3	40
4	50
5	

```
a1 = new int[5];
```

2D dimensional:

```
int a1[][] = new int[5][5];
```

```
<data type> <arrayname>[][] = new <data type>[row][col];
```

row: no of arrays in the 2D array

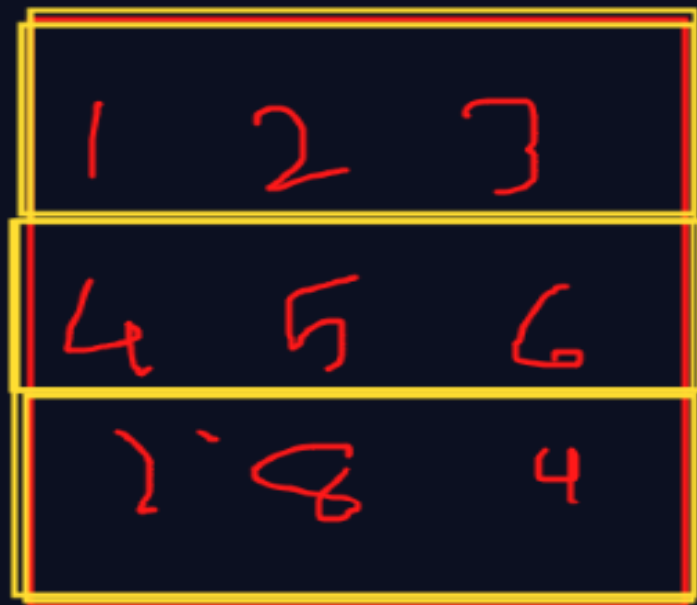
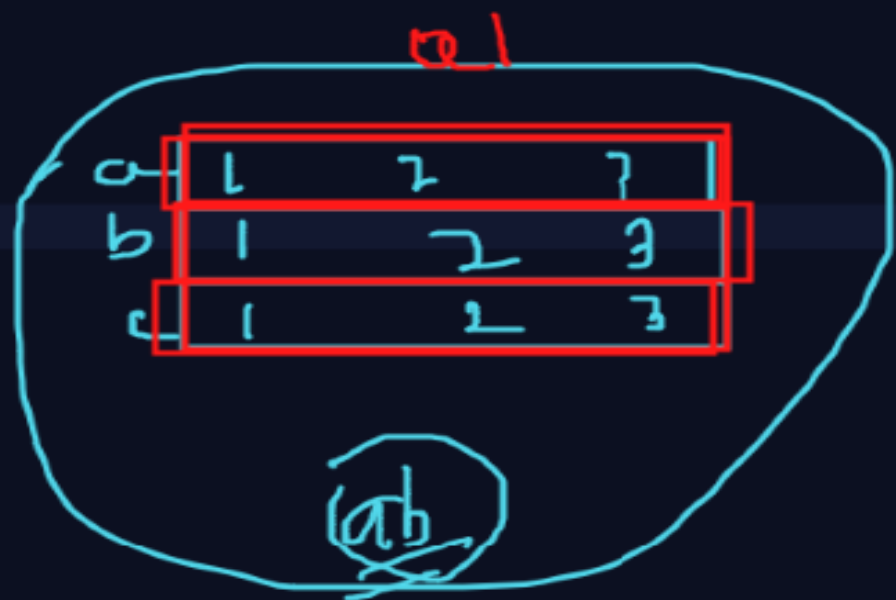
col: no of elements present in each array

a11

0	10	✓ ⁹
1	20	✓ ^u
2	30	✓ ^e
3	40	5
4	50	λ
5		

a1				
a2				
a3				
a4				

```
class Array5{  
    public static void main(String args[]){  
        int a[]={1,2,3};  
        int b[]={4,5,6};  
        int c[]={7,8,9};  
  
        int a1[][] = new int[3][];  
  
        a1[0] = a;  
        a1[1] = b;  
        a1[2] = c;  
  
        //for each loop  
        for(char x:a1)  
        {  
            System.out.println(x);  
        }  
    }  
}
```



```
class Array5{
```

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

```
        int a[]={1,2,3};
```

```
        int b[]={4,5,6};
```

```
        int c[]={7,8,9};
```

```
        int a1[][] = new int[3][];
```

```
        a1[0] = a;
```

```
        a1[1] = b;
```

```
        a1[2] = c;
```

```
        for(int i=0;i<a1.length;i++) //Rows
```

```
        {
```

```
            for(int j=0;j<a1[i].length;j++) //cols
```

```
            {
```

```
                System.out.println(" "+a1[i][j]);
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

class Array6{

public static void main(String args[]){

int a[]={1,2,3};

int b[]={4,5,6};

int c[]={7,8,9};

1	2	3
4	5	6
7	8	9

int al[][] = new int[3][];

al[0] = a;

al[1] = b;

al[2] = c;

```
for(int i=0;i<al.length;i++)//Rows
{
    for(int j=0;j<al[i].length;j++)//cols
    {
        System.out.print(" "+al[i][j]);
    }
    System.out.println();
}
```

}

Command Prompt

4

5

6

7

8

9

C:\Test>javac Array6.java

C:\Test>java Array6

1 2 3

4 5 6

7 8 9

C:\Test>