



JAVA

Concepts of Programming Day 5: March 2022

Kiran Waghmare



java

```

{
double sal;
String name;
static float da= 0.1f;
static float in=0.07f;
void getdata(double sal, String name)
{
this.sal=sal;
this.name= name;
//double d=(sal*da);
//double i=(sal*in); ( it give error )(reason-)
}
void setdata()
{
System.out.println("employee name "+name);
System.out.println("basic salary "+sal);
double d=(sal*da);
double i=(sal*in);
double total=(sal+d+i);
System.out.println("d.a = "+d);
System.out.println("increment = "+i);
System.out.println("total salary = "+total);
}
}
class run
{
public static void main(String [] args)
{
gross_salery gs = new gross_salery();
Scanner sc = new Scanner(System.in);

```

```

}
void setdata()
{
System.out.println("employee name "+name);
System.out.println("basic salary "+sal);
double d=(sal*da);
double i=(sal*in);
double total=(sal+d+i);
System.out.println("d.a = "+d);
System.out.println("increment = "+i);
System.out.println("total salary = "+total);
}
}
class run
{
public static void main(String [] args)
{
gross_salery gs = new gross_salery();
Scanner sc = new Scanner(System.in);
System.out.println("plss enter the employee name");
gs.name = sc.nextLine();
gs.getdata(500000,gs.name);
gs.setdata();
}
}

```

Access Specifier/Modifiers:

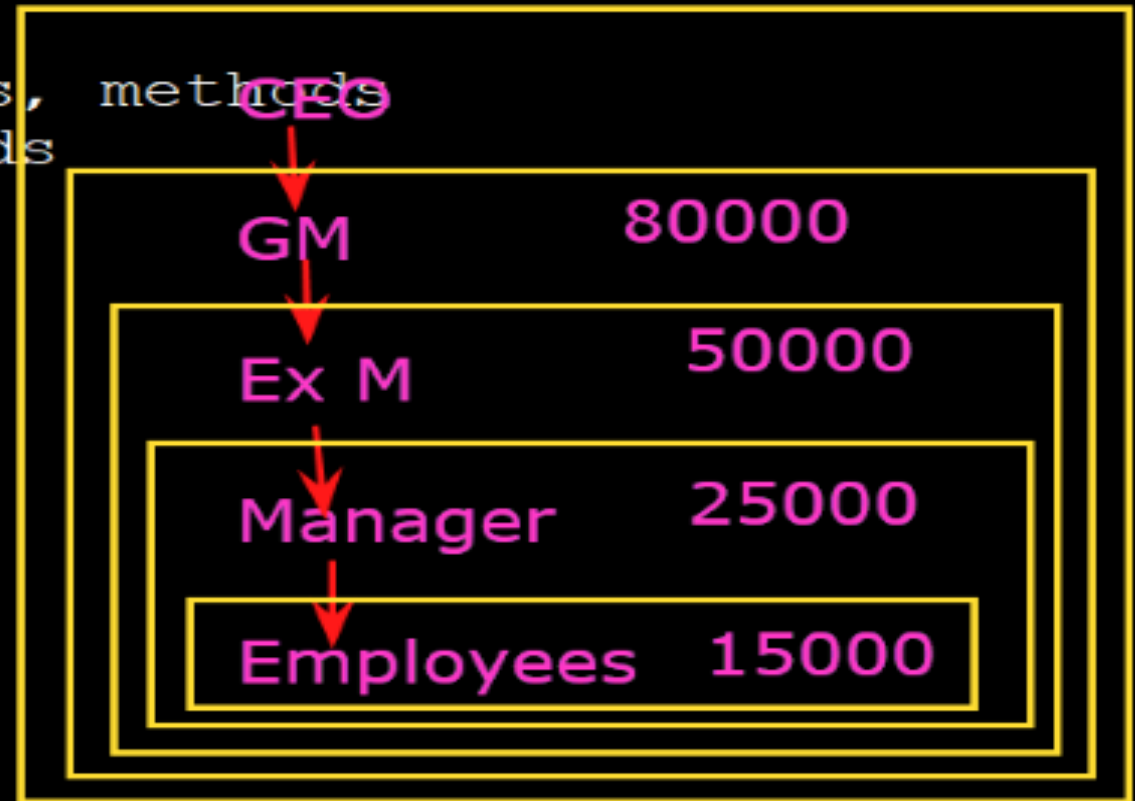
- used to set accessibility
- retricts access of datamembers, methods
- class, block, variables, methods

Access modifiers:

- default
- Public
- Private
- Protected

Non-Access Modifiers:

- static
- final
- abstract
- synchronised & volatile



Access Specifier/Modifiers:

- used to set accessibility
- retricts access of datamembers, methods
- class, block, variables, methods

Access modifiers:

-default:

-VISIBILITY TO THE PACKAGE

-Public:

-VISIBILITY TO THE WORLD

-Private:

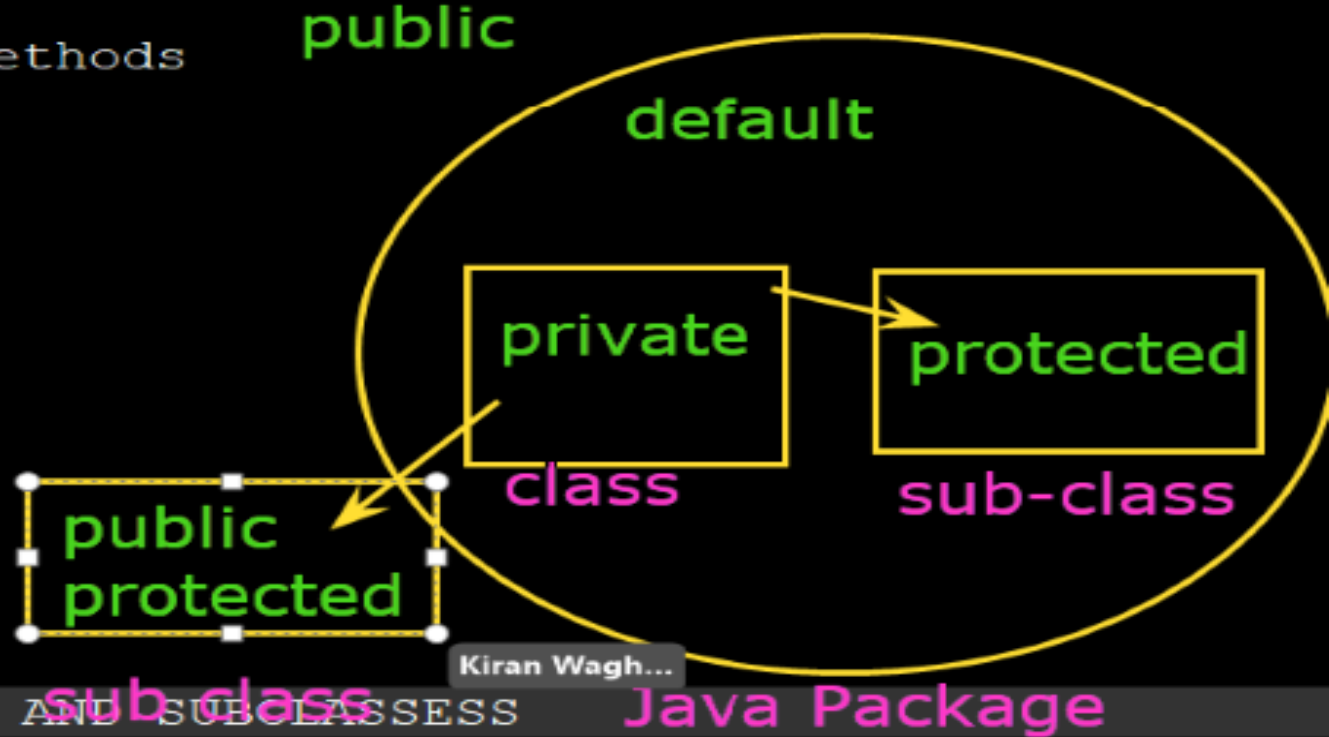
-VISIBILITY TO ONLY CLASS

-Protected:

-VISIBILITY TO THE PACKAGE AND SUBCLASSESS

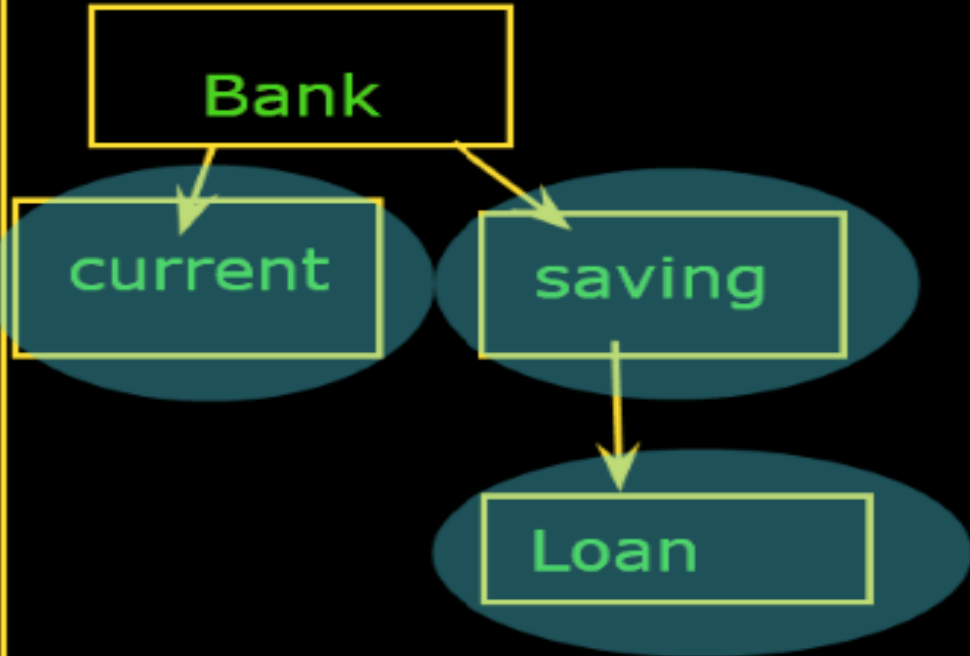
Non-Access Modifiers:

- static
- final
- abstract
- synchronised & volatile





Inheritance



public

default

public
protected

sub class

private

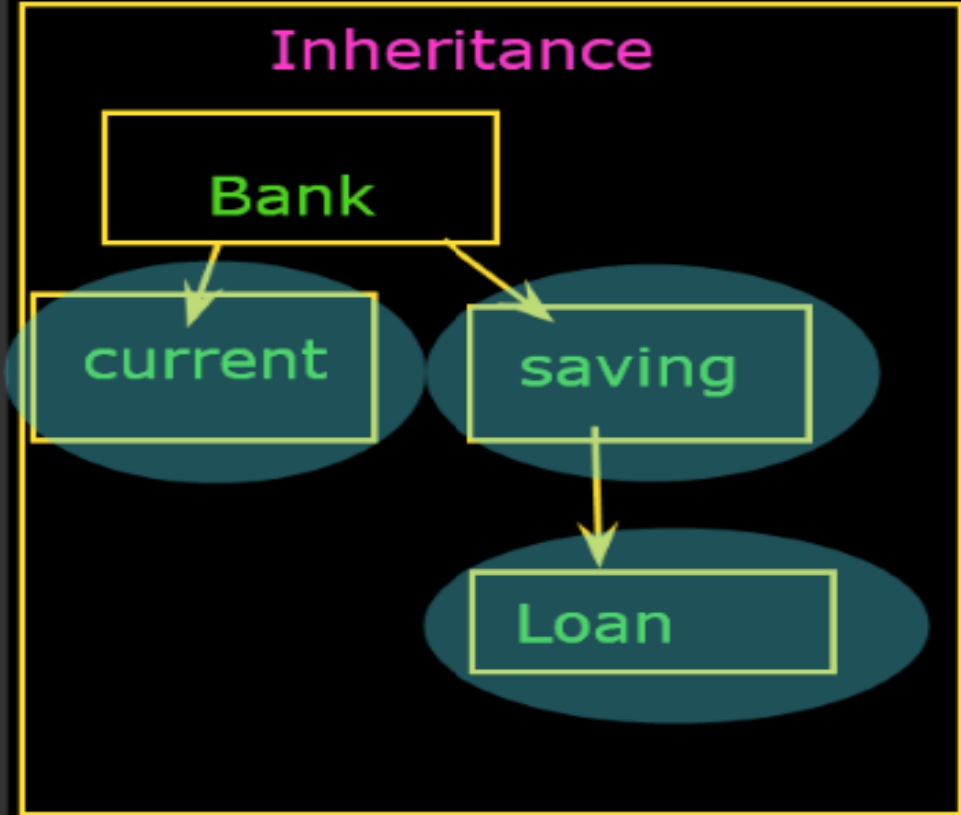
class

protected

sub-class

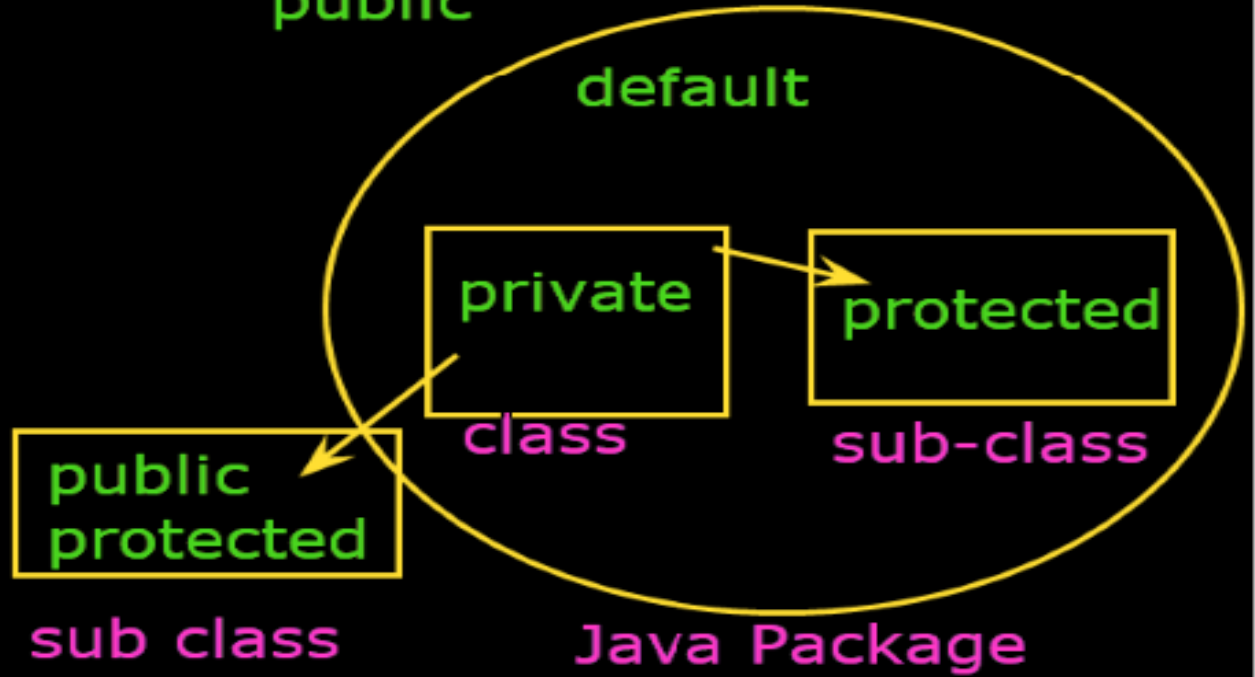
Java Package

Inheritance



public

default



Array Declare:

1.new operator

```
-e.g.int a1[] = new int[5];
```

2.array initializer

```
-int []a1={1, 3, 5, 7, 9};
```

```
-int a1[]={1, 3, 5, 7, 9};
```



Examples:

```
int a1[];
```

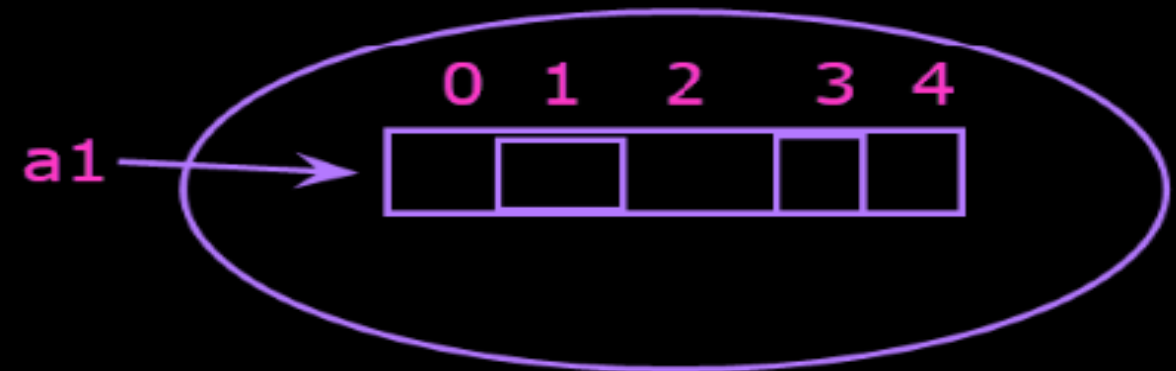
```
int []a1;
```

```
int a1[],b,c;
```

```
int a1[],b[],c[];
```

```
a1 [] a,b,c;
```

```
int a,b,c,b;
```



$a1[2]=5;$

Array Declare:

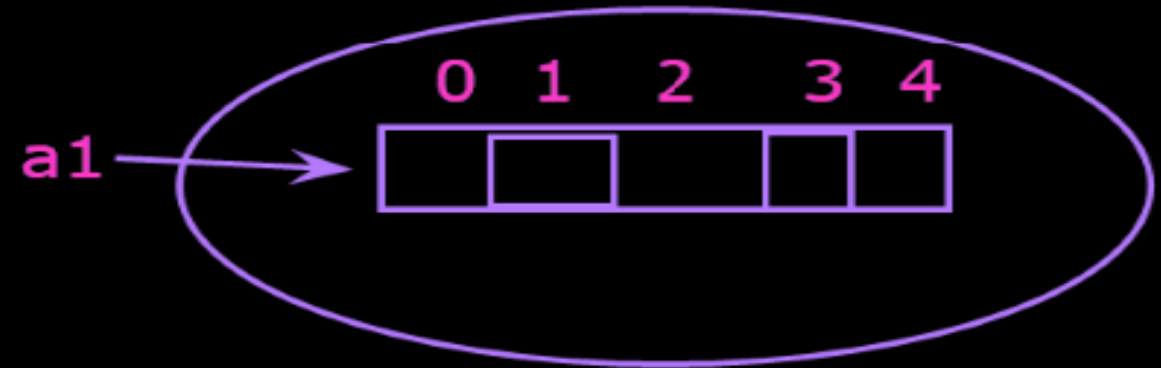
1.new operator

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

2.array initializer

-int []a1={1, 3, 5, 7, 9};

-int a1[]={1, 3, 5, 7, 9};



a1[2]=5;

Examples:

```
int a1[];  
int []a1;  
int a1[],b,c;  
int a1[],b[],c[];  
a1 [] a,b,c;
```

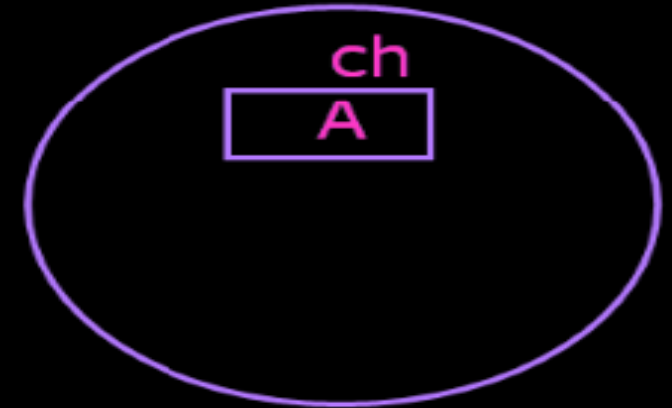
```
int a,b,c,b;
```

Array

```
import java.util.*;
class Array6
{
```

```
    public static void main(String args[])
    {
        char ch[] = {'P','R','A','T','I','K'};

        //char ch[] = new char[];
        for(char x: ch){
            System.out.println("Array element = "+x);
        }
    }
}
```



Command Prompt

```
C:\CDAC22>javac Array6.java
```

```
C:\CDAC22>java Array6
```

```
Array element = P
```

```
Array element = R
```

```
Array element = A
```

```
Array element = T
```

```
Array element = I
```

```
Array element = K
```

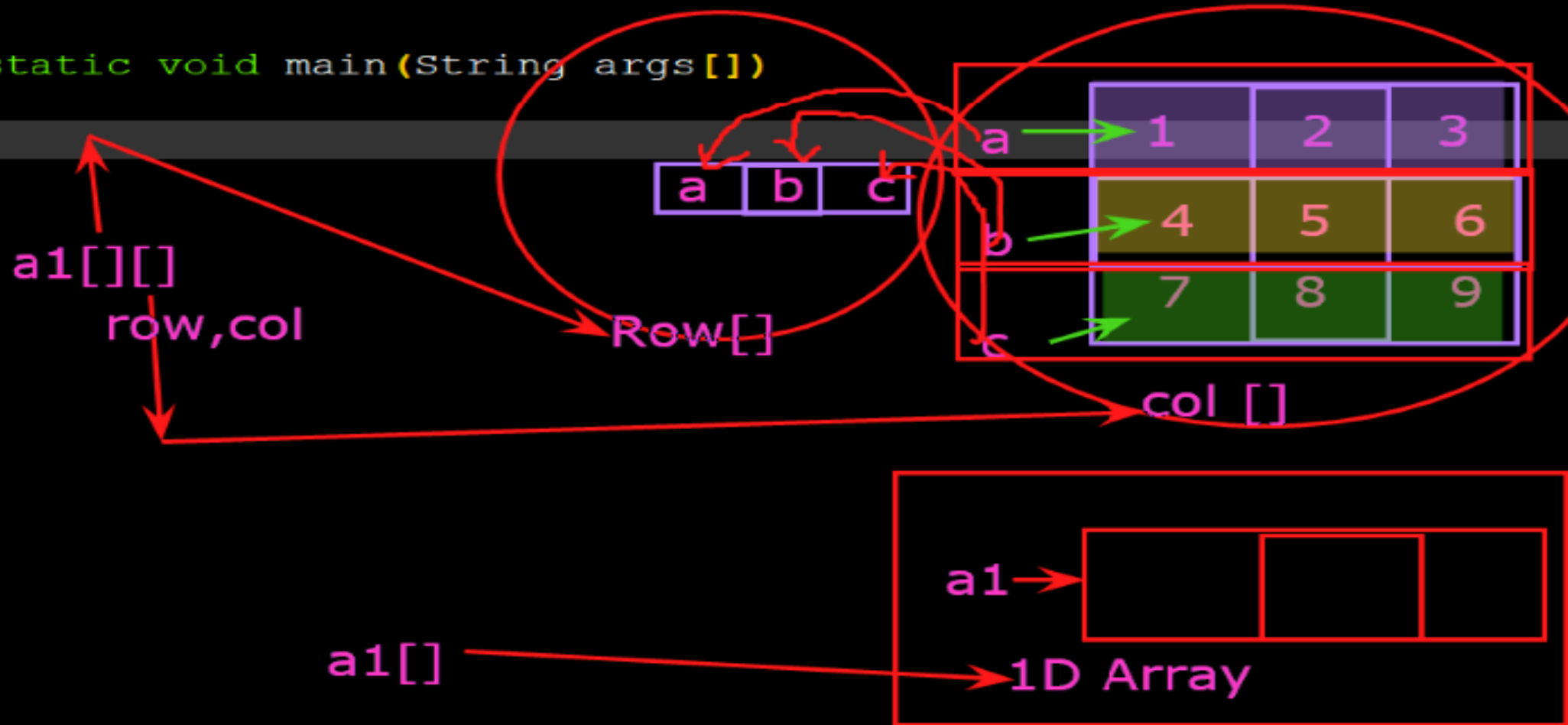
```
C:\CDAC22>
```



```

import java.util.*;
class Array7
{
    public static void main(String args[])
    {
    }
}

```



```
import java.util.*;
class Array7
{
    public static void main(String args[])
    {
    }
}
```

Mouse

Select

Text

Draw

Stamp

Spotlight

Eraser

Format

Undo

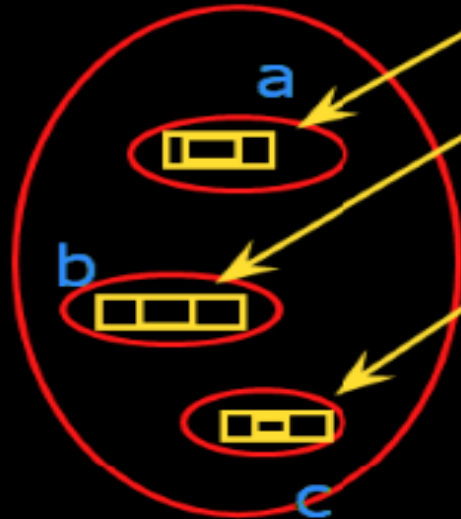
Who can see what you share here? Recording On

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

a1[][]
row,col

Row[]

col []



a1[]

a1 →

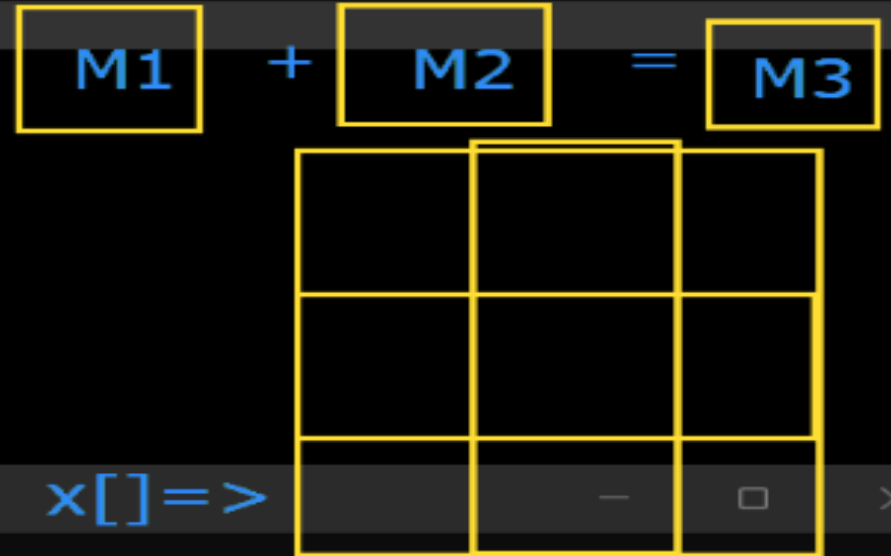
1D Array



```
import java.util.*;
class Array10
{
```

```
public static void main(String args[]){
    Scanner sc = new Scanner(System.in);
    int a1[][]=new int[3][3];
    int b1[][]=new int[3][3];

    //Matrix Input 1
    for(int i=0;i<a1.length;i++){//Row
```



Array elements:2
Array elements:2
Array elements:2

1 2 1
1 1 1
1 1 1
→ M1

2 2 2
2 2 2
2 2 2
→ M2

=

3 4 3
3 3 3
3 3 3
M3

C:\CDAC22>2

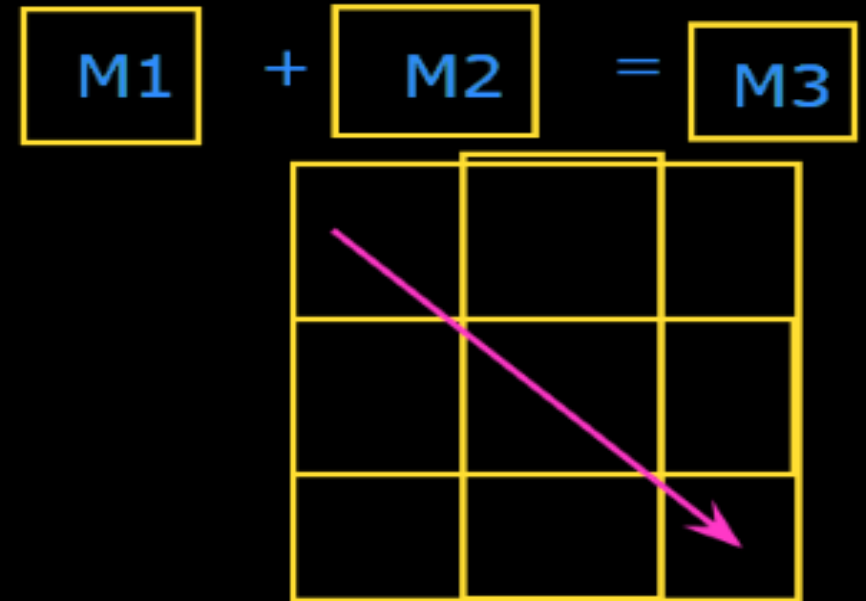
```
System.out.print("Array elements:");
b1[i][j]=sc.nextInt();//input
```


e.g

```
for(int x : a1)
{
    SOP();
}
```

Matrix programs:

- 1.Addition of 2 matrix
- 2.Substraction of 2 matrix
- 3.Multiplication of 2 matrix
4. Transpose of a matrix
- 5.Sum of all elements of matrix
- 6.print diagonal element of the matrix



```
char ch[] = {'S', 'a', 'u', 'r', 'a', 'b', 'h'};  
String str = new String("Rohit");  
String str1 = "cdac mumbai"
```

Ways to create a string:

1. String literal

eg: String str1 = "cdac mumbai"

2. using new keyword

String str = new String("Rohit");

String s1 = "Hello";

String s2 = "India";

String constant pool:

- String objects are stored in a special memory area
- Heap memory
- memory efficient programming
-



```
import java.util.*;  
class S1  
{
```

```
    public static void main(String args[]){  
        String str = "Hello";  
        String s1 =str.concat(" India");//Hello India  
        str=str.concat(" India");//Hello India  
        //concat: add the string at the end  
        System.out.println(str);  
        System.out.println(s1);  
        System.out.println(str);  
    }
```

Command Prompt

```
Hello  
Hello India
```

```
C:\CDAC22>javac S1.java
```

```
C:\CDAC22>javac S1.java
```

```
C:\CDAC22>java S1  
Hello India  
Hello India  
Hello India
```

```
C:\CDAC22>
```



```
import java.util.*;
```

```
class S1
```

```
{
```

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

```
        String str = "Hello";
```

```
        String s1 =str.concat(" India");//Hello India
```

```
        str=str.concat(" India");//Hello India
```

```
        //concat: add the string at the end
```

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

```
        System.out.println(s1);
```

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

```
    }
```

```
}
```

Mouse

Select

Text

Draw

Stamp

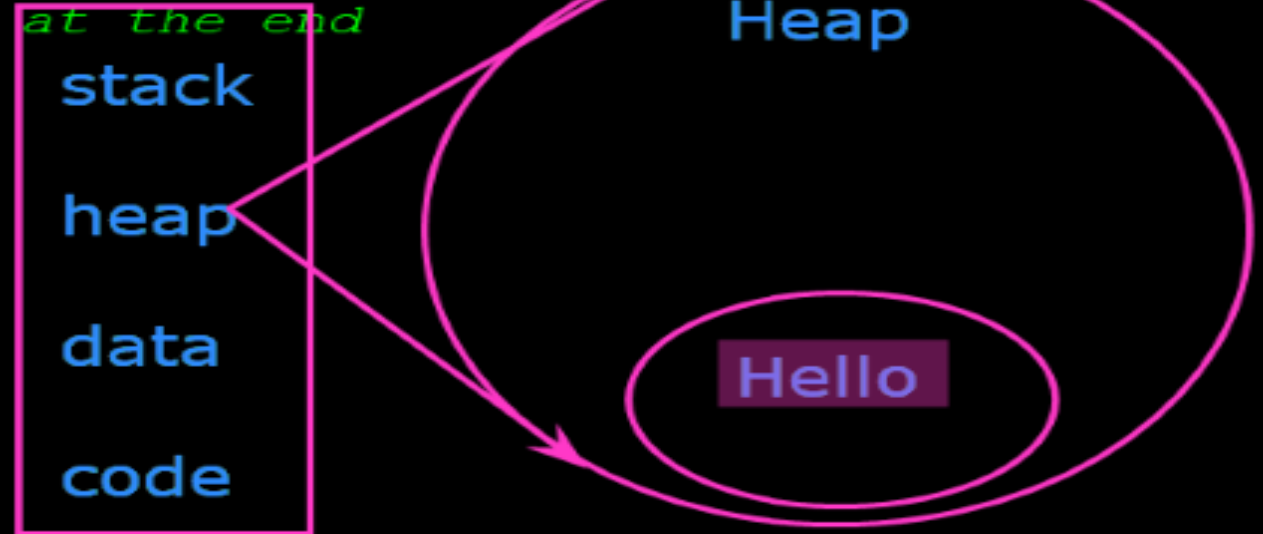
Spotlight

Eraser

Format

Undo

Who can see what you share here? Recording On



- String objects are stored in
- Heap memory
- memory efficient programming
 - Use less memory

```
String str = "Hello";  
String s1 = "India"
```

```
String str = "Hello";  
String s1 = new String("India");
```

