Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

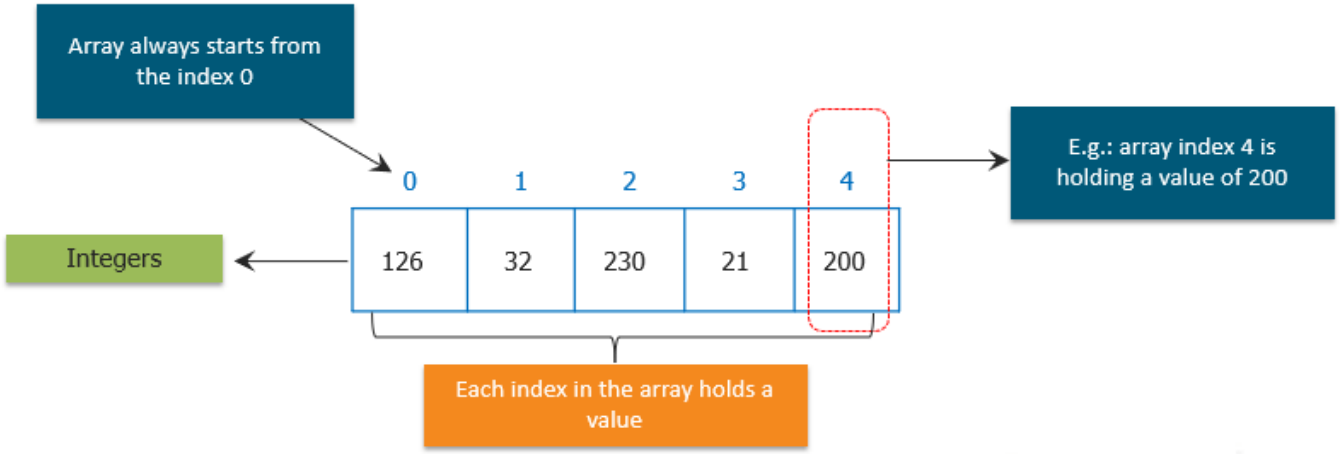
Academic Year: 2022-2023

Name – Prerna Sunil Jadhav **SAP ID - 60004220127**

Experiment No - 03

AIM: TO IMPLEMENT ARRAYS

THEORY:



An array is a collection of similar types of data.

For example, if we want to store the names of 100 people then we can create an array of the string type that can store 100 names.

```
String[] array = new String[100];
```

Here, the above array cannot store more than 100 names. The number of values in a Java array is always fixed.
In Java, here is how we can declare an array.

```
dataType[] arrayName;
```

datatype - it can be primitive data types like int, char, double, byte, etc. or Java objects
arrayName - it is an identifier

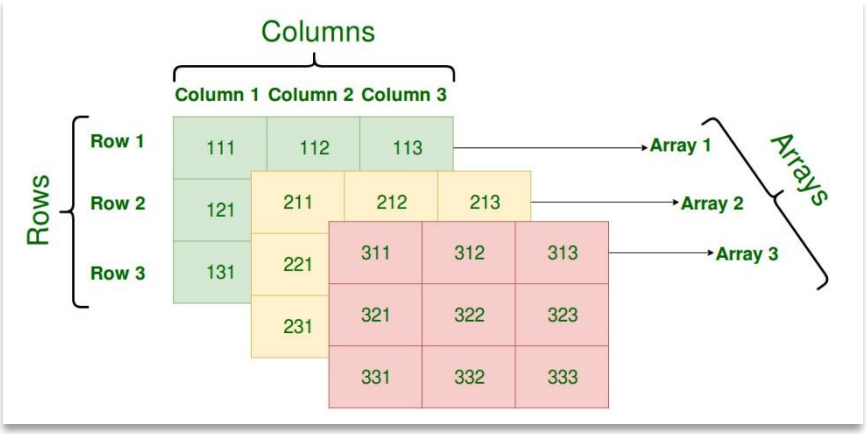
For example,

```
double[] data;
```

Here, data is an array that can hold values of type double.

A multidimensional array is an array of arrays. Each element of a multidimensional array is an array itself.

```
int[][] a = new int[3][4];
```



PROGRAM 1: Write A Program to find whether the entered 4 digit number is vampire or not. Combination of digits from this number forms 2 digit number. When they are multiplied by each other we get the original number. (1260=21*60, 1395=15*93, 1530=30*51)

CODE:

Code1_Vampire.java

Exp3 > Code1_Vampire.java > Code1_Vampire > main(String[])

```
1   package Exp3;
2
3   import java.util.*;
4
5   class Code1_Vampire
6   {
        Run | Debug
7       public static void main(String args[])
8       {
9           System.out.println(x: "Prerna Sunil Jadhav - 60004220127");
10          Scanner sc=new Scanner(System.in);
11          int v=0;
12          long n,i,c=0,f=0,j=0,x,k,p,c1,c2,b[]=new long[100];
13          System.out.println(x: "Enter a number:");
14          n=sc.nextLong();
15          sc.close();
16          //Counting the number of digits
17          for(i=n;i>0;i/=10)
18              c++;
19          if(c%2!=0)
20              f=1;
21          else
22          {
23              for(i=(int)Math.pow(a: 10,c/2-1); i<(int)Math.pow(a: 10,c/2);  i++)
24              {
25                  if(n%i==0)  //i is one factor
26                  {
27                      j=n/i;   //j is another factor
28                      if(!(j>=Math.pow(a: 10,c/2-1) && j<Math.pow(a: 10,c/2)))
29                          f=1;
30
```

Code1_Vampire.java

Exp3 > Code1_Vampire.java > Code1_Vampire > main(String[])

```
                    x=i*(int)Math.pow(a: 10,c/2)+j;

                    if(i%10==0 && j%10==0)
                    {
                        f=1;
                        break;
                    }
                    else //Check all digits are distinct or not
                    {
                        f=0;
                        for(k=n;k>0;k/=10)
                        {
                            c1=c2=0;
                            for(p=n;p>0;p/=10)
                            {
                                if(k%10==p%10)
                                    c1++;
                            }
                            for(p=x;p>0;p/=10)
                            {
                                if(k%10==p%10)
                                    c2++;
                            }
                            if(c1!=c2)
                            {
                                f=1;
                                break;
                             }
                        }
                        if(f==0)
                        {
                            int fl=0;
                            for(int ii=0;ii<v;ii++)
                                if(b[ii]==i || b[ii]==j)
                                    fl=1;
                            if(fl==0)
                                b[v++]=i;

                         }
                    }
                }
            }

        }
```

```
        if(v==0)
            System.out.println(x: "Not a vampire number");
        else
        {
            System.out.println(x: "Vampire number fangs are: ");
            for(int ii=0;ii<v;ii++)
                System.out.println(b[ii]+"\t"+(n/b[ii]));
        }
    }
}
```

OUTPUT:

```
Prerna Sunil Jadhav - 60004220127
Enter a number:
1260
Vampire number fangs are:
21      60
```

```
Prerna Sunil Jadhav - 60004220127
Enter a number:
1593
Not a vampire number
```

PROGRAM 2: WAP to display the following using irregular arrays

2 3
4 5 6

CODE:

Code2_IrregularArray.java

```
Exp3 > Code2_IrregularArray.java > ...
package Exp3;

public class Code2_IrregularArray {
    Run | Debug
    public static void main(String[] args) {
        System.out.println(x: "Prerna Sunil Jadhav - 60004220127");
        // Declaring 2-D array with 2 rows
        int arr[][] = new int[2][];
        arr[0] = new int[2];    // First row has 3 columns
        arr[1] = new int[3];    // Second row has 2 columns
        int count = 2;
        for (int i = 0; i < arr.length; i++)
            for (int j = 0; j < arr[i].length; j++)
                arr[i][j] = count++;                    // Initializing array
        // Displaying the values of 2D Jagged array
        for (int i = 0; i < arr.length; i++) {
            for (int j = 0; j < arr[i].length; j++)
                System.out.print(arr[i][j] + " ");
            System.out.println();
        }
    }
}
```

OUTPUT:

```
Prerna Sunil Jadhav - 60004220127
2 3
4 5 6
```

PROGRAM3: WAP that queries a user for no. of rows & columns representing students & their marks. Reads data row by row & displays the data in the tabular form along with the row & columns & grand total.

CODE:

Code3_2DArray.java

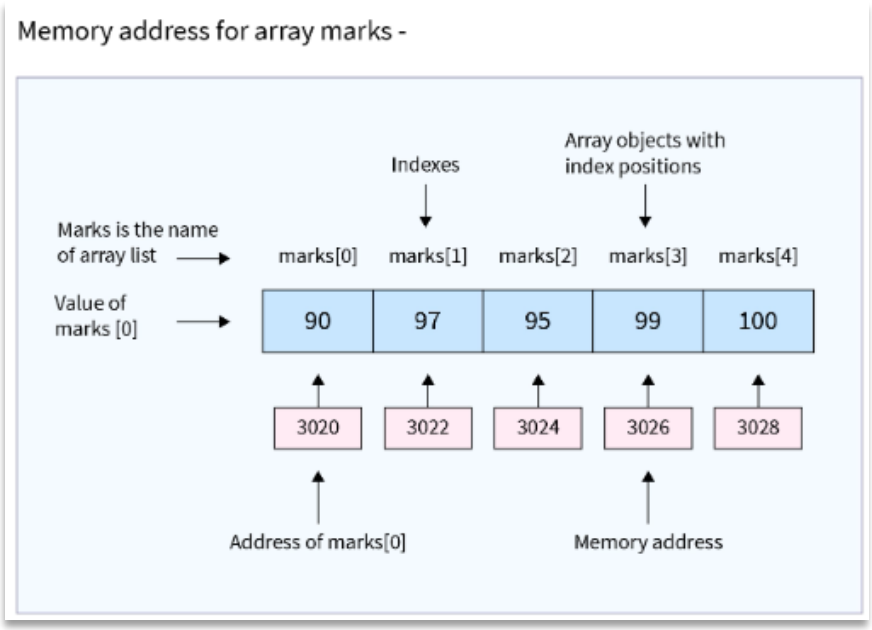
Exp3 > Code3_2DArray.java > ...

```java
package Exp3;
import java.util.Scanner;
public class Code3_2DArray {
    Run | Debug
    public static void main(String[] args) {
        System.out.println(x: "Prerna Sunil Jadhav - 60004220127");
        Scanner sc = new Scanner(System.in);
        System.out.print(s: "Enter Number of Students: ");
        int row = sc.nextInt();
        System.out.print(s: "Enter Number of Subjects: ");
        int col = sc.nextInt();
        int student[][] = new int[row][col];
        for (int i = 0; i<row; i++){
            System.out.println("Enter Marks for Student "+(i+1)+" :");
            for (int j = 0; j<col; j++){
                System.out.print("Enter Marks for Subject "+(j+1)+" : ");
                student[i][j] = sc.nextInt();
            }
        }
        for (int i = 0; i<row; i++){
            int sum = 0;
            for (int j = 0; j<col; j++)
                System.out.print(student[i][j]+"\t");
            for (int k = 0; k<col; k++)
                sum += student[i][k];
            System.out.println(" | "+sum);
        }
        System.out.println(x: "----------------------------");
        int total =0;
        for (int i = 0; i<col; i++){
            int sum=0;
            for(int j = 0; j<row; j++)
                sum += student[j][i];
            System.out.print(sum + "\t");
            total += sum;
        }
        System.out.println(" | "+total);
        sc.close();
    }
}
```

OUTPUT:

```
Prerna Sunil Jadhav - 60004220127
Enter Number of Students: 2
Enter Number of Subjects: 3
Enter Marks for Student 1 :
Enter Marks for Subject 1 : 1
Enter Marks for Subject 2 : 3
Enter Marks for Subject 3 : 6
Enter Marks for Student 2 :
Enter Marks for Subject 1 : 7
Enter Marks for Subject 2 : 9
Enter Marks for Subject 3 : 8
1       3       6       | 10
7       9       8       | 24
---------------------------
8       12      14      | 34
```

CONCLUSION:



- Array indices always start from 0. That is, the first element of an array is at index 0.
- If the size of an array is n, then the last element of the array will be at index n-1.
- Array in Java is a non-primitive data type used to store multiple values of the same data type.
- Elements of the array can be accessed by the index ranging from 0 to array length - 1.
- We can use for loop and for-each loop for looping through all the array elements.
- There are two arrays in Java: Single and Multi-dimensional, where single has one dimension, and multi includes 2D, 3D, and nD dimensions.
- Likewise primitive data types, we can also implement an array of objects in Java.