Arrays

An array is an object that stores a list of items continuously of same data type

There are

1. **Single Dimensional** and
2. **Multi Dimensional** arrays.

5.1 Single Dimension Array

Single Dimenstion Array stores the element in a linear manner

Declaration

5.1.1 Array Initialization

Syntax `data-type array-name[] = {value1,value2,.....};`

Example `int a[]= {10,20,50,67,77}`

5.1.2 Array with size

Syntax `datatype array-name [] = new datatype[size];`

Example `int a[] = new int[5]; or int[] a = new int[5];`

(or)

```
int a[];
-------------------;
-------------------;
a[]= new int[5];
```

The **new** keyword creates an instance of an array

Program

```
// program to calculate sum of an array
import java.util.Scanner;

class SingleDimenstionArrayDemo
{
      public static void main(String arg[])
      {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter the array size ");

      int size=s.nextInt();

      int a[] = new int[size];
      int sum=0;
      for(int i=0;i<n;i++)
      {
            a[i]=s.nextInt();
            sum=sum+a[i];
      }
      System.out.println("Sum of an Array ="+sum);
      }
```

```
}
```

Output

```
Enter the Array size
10 15 30 40 25
Sum of an Array 120
```

5.2 Two Dimension Array

Two dimensional arrays help you to manipulate table of data. It represents data to be stored in the form of rows and columns

Declaration

Syntax

```
datatype array_name[][] = new datatype[rowsize][col.size];
```

Example

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

Program

```
//To calculate sum of diagonal

import java.util.Scanner;

class TwoDimenstionArrayDemo
{
public static void main (String arg []) throws IOException
{
      Scanner s = new Scanner(System.in);
      System.out.println("Enter the row size ");

      int row=s.nextInt();

      System.out.println("Enter the column size ");

      int col= s.nextInt();

      int a[][]= new int [row][col];

      int sum =0;
      for (int i=0;i<row;i++)
      {
            for (int j=0;j<col;j++)
            {
                  a[i][j]= s.nextInt();
            }
      }

      for (int i=0;i<row;i++)
      {
            for (int j=0;j<col;j++)
            {
                  System.out.print(a[i][j]+"\t");
                  if(i==j)
                  sum = sum +a[i][j];
            }
            System.out.println();
      }
      System.out.println("Sum of diagonal is " +sum);
}
}
```

Output

```
Enter the row size 3
Enter the column size 3
```

```
1 2 3
4 5 6
7 8 9
Sum of diagonal is 15
```

6. Methods (Functions)

Methods are an integral and important part of Object programming language.

There are several advantages of Methods

1. The individual methods (blocks) will be easier to code, test and debug.
2. The same block can be reused in different parts of the programs.
3. Commonly used blocked can be collected together in a library so that they can be used by any other program

You have already come across several such methods in earlier chapters.

Example **`System.out.println(), readLine(), Integer.parseInt();`**

6.1 Method Declaration

Syntax

```
access-specifier  [static]   return_type methodName(argument list)
{
      Local variable declaration;
      Executable part;
      return (expression);
}
```

6.2 Invoking a Method

Syntax `methodName(parameter list);`

6.2.1 Types of Method Declaration

1. Method **with arguments** and **with return** type
2. Method **without arguments** and **with return** type
3. Method **with arguments** and **without return** type
4. Method **without arguments** and **without return** type

Type 1

Example Method **with arguments and with return type**

```
//Program to calcualte sum of two numbers using method

    class Method1
      {
            public static int sum(int a, int b)
            {
                  int c;
                  c=a+b;
                  return(c);
            }
            public static void main(String args[])
            {
```

```
                int x=Integer.parseInt(args[0]);
                int y=Integer.parseInt(args[1]);
                int z=sum(x,y);
                System.out.println("Sum of two numbers is = "+z);

        }
}
```

Output

```
c:\>java Method1 10 20
Sum of two numbers is 30
```

Type 2

Example Method **without arguments and with return type**

```
//Program to find biggest of two numbers

      class Method2
      {
            public static int biggest()
            {
            int a=10,b=20;
                  if(a>b)
                  return a;
                  else
                  return b;
            }
            public static void main(String args[])
            {
                  int big;
                  big=biggest();
                  System.out.println("Biggest of two numbers is = "+big)
            }
      }
```

Output

```
Biggest of two numbers is 20
```

Type 3

Example Method **with argument and without return type**

```
//Program for swapping two numbers


      class Method3
      {
            static void swap(int x, int y)
            {
                  int t;
                  t=x;
                  x=y;
                  y=t;
                  System.out.println("X = "+ x +"Y = "+ y);
            }
            public static void main(String args[])
            {
                  int a=10, b=20;
                  swap(a,b) //calling of swap function
                  System.out.println("End of main function");
```

```
        }
    }
```

Output

```
X = 20 Y = 10
End of main Function
```

Type 4

Example Method **without argument and without return type:**

```
//Program to swap of two numbers

class Method4
{
    static void swap()
    {
        int a=10, b=20,t;
        t=a;
        a=b;
        b=t;
        System.out.println("A = "+ a +"B = "+ b);
    }
    public static void main(String args[])
    {
        swap() //calling of swap function
        System.out.println("End of main function");
    }
}
```

Output

```
A = 20 B = 10
End of main function
```