Arrays in Java

Assignment-9

1.What do you mean by an Array ?

Ans: An array is a data structure that holds a collection of elements, which can be of any data type (e.g. integers, strings, objects, etc.). The elements are stored in contiguous memory locations, and can be accessed by a zero-based index. Arrays are often used to store multiple values of the same data type, and provide a way to organize and manipulate collections of data.

For example, you can create an array to store a list of numbers: [1, 2, 3, 4, 5]. You can then access each element in the array by its index, such as array[0] to access the first element (1), array[1] to access the second element (2), and so on.

2.How to create an Array ?

Ans: There are several ways to create an array in Java. Here are a few examples:

1. Using the **array initializer**

```
int[] numbers = {1, 2, 3, 4, 5};
```

2. Using the **new operator**:

```
int[] numbers = new int[]{1, 2, 3, 4, 5};
```

3. Specifying the size and then initializing the elements:

```
int[] numbers = new int[5];
numbers[0] = 1;
numbers[1] = 2;
...
numbers[4] = 5;
```

Note that the size of an array in Java is fixed and cannot be changed after it is created.

3.Can we change the size of an array at run time ?

Ans: No, the size of an array in Java is fixed and cannot be changed once it has been created. If you need to change the size of an array, you can create a new array with a different size,

copy the elements from the original array to the new array, and then use the new array instead of the original one.

Here's an example of how you could increase the size of an array in Java:

```
int[] numbers = {1, 2, 3, 4, 5};


int[] newNumbers = new int[numbers.length + 1];

System.arraycopy(numbers, 0, newNumbers, 0, numbers.length);


numbers = newNumbers;
```

4.Can you declare an array without assigning the size of an array ?

Ans: No, it's not possible to declare an array without specifying its size in Java. When you declare an array in Java, you need to specify the size of the array, which determines the number of elements the array can hold.

Here's an example of how to declare an array in Java with a size of 10:

```
int[] numbers = new int[10];
```

In this example, the `numbers` array has been declared with a size of 10, meaning it can hold 10 elements.

5.What is the default value of Array ?

Ans: The default value of an array in Java depends on the data type of the elements the array is storing. The default values for the primitive data types are:

- `boolean`: `false`
- `char`: `'¥u0000'` (the null character)
- `byte`, `short`, `int`, `long`: `0`
- `float`, `double`: `0.0`

For reference types (e.g., objects and arrays), the default value is `null`.

For example, if you declare an array of integers like this:

```
int[] numbers = new int[10];
```

The default value of each element in the `numbers` array will be `0`

6.What is a 1 D array with an Example ?

Ans: A one-dimensional (1D) array is a linear data structure that holds a collection of elements, where each element is identified by an index. In other words, a 1D array is a sequence of elements of the same data type stored in contiguous memory locations.

Here's an example of how to declare and initialize a 1D array in Java:

int[] numbers = new int[] {1, 2, 3, 4, 5};

n this example, the **numbers** array is a 1D array of type **int** that holds 5 elements with values **1**, **2**, **3**, **4**, and **5**. The elements in a 1D array can be accessed using their index, which starts from **0** and goes up to **n-1**, where **n** is the size of the array. For example, to access the first element in the **numbers** array, you would use the following code:

System.out.println(numbers[0]); // outputs 1

7.What a Program on a 2D array ?

Ans: Addtion of two matrix by using methods are:

```
import java.util.Scanner;

public class Two_D_array_Addition_of_two_matrix_using_function {

   static void print_matrix(int matrix[][]){

      for(int i=0;i<matrix.length;i++){

         for(int j=0;j<matrix[i].length;j++){

            System.out.print(" "+matrix[i][j]);
         }
         System.out.println();
      }
   }

   static void add(int array1[][],int r1,int c1,int array2[][],int r2,int c2){

      if(r1!=r2 || c1!=c2){

         System.out.println("Invalid input: addition is not possible:");
         return;
      }

      int sum[][]=new int[r1][c1];

      for(int i=0;i<r1;i++){  // Row...

         for(int j=0;j<c1;j++){  // column....

            sum[i][j]=array1[i][j]+array2[i][j];
         }
```

```
            System.out.println();
        }
        System.out.println(" Addition of two matrix :");
        print_matrix(sum);
    }

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the value of Row-1 and column-1 of Matrix-1");
        int r1=sc.nextInt();
        int c1=sc.nextInt();

        int array1[][]=new int[r1][c1];

        System.out.println("Enter the value of Matrix-1");

        for(int i=0;i<r1;i++){  // Row.....

            for(int j=0;j<c1;j++){   // column...

                array1[i][j]=sc.nextInt();
            }
        }

        System.out.println("Enter the value of Row-2 and column-2");
        int r2=sc.nextInt();
        int c2=sc.nextInt();

        int array2[][]=new int[r2][c2];

        System.out.println("Enter the value of Matrix-2");

        for(int i=0;i<r2;i++){   // Row....

            for(int j=0;j<c2;j++){  // column..

                array2[i][j]=sc.nextInt();
            }
        }

        System.out.println("Value of Matrix-1:");
        print_matrix(array1);

        System.out.println("value of Matrix-2:");
        print_matrix(array2);

        System.out.println("Sum of Two matrix-1 and matrix-2:");
        add(array1,r1,c1,array2,r2,c2);
    }
}
```

Output:

```
Enter the value of Row-1 and column-1 of Matrix-1
3 3
Enter the value of Matrix-1
1 2 3
4 5 6
7 8 9
Enter the value of Row-2 and column-2
3 3
Enter the value of Matrix-2
1 2 3
4 5 6
7 8 9
Value of Matrix-1:
1 2 3
4 5 6
7 8 9
value of Matrix-2:
1 2 3
4 5 6
7 8 9
Addition of two matrix :
2 4 6
8 10 12
14 16 18

Process finished with exit code 0
```