20/06/23

Working with arrays:

Working with arrays and strings in java involves operations like creating, manipulating and accessing elements.

1. Declaring & Initialization of Arrays:

```
// Declare an array
int [] numbers;
// initialize the array with values
numbers = new int[]{1,2,3,4,5};
```

2. Accessing array elements

```
int FirstNumber = numbers[0]; // Access first element of array
int length = numbers.length;
//Getting the length of array
```

3. Modify the array element

```
numbers[0] = 10; // Modify value at index '0'
```

4. Iterating over arrays

```
int i;
for (i=0; i< numbers.length; i++)
{
    int value = numbers[i];
}
```

// Enhance for loop(for-each loop)

```
for (int value: numbers){
        // commands
}
```

Predefined methods for arrays:

1. Arrays.toString(array): converts an array to a string representation.

```
int [] numbers = {1,2,3,4,5};
String array_string = Arrays.toString(numbers);
S.O.P(array_string);
```

O/p: [1,2,3,4,5]

2. Arrays.sort(array): sorts an array in ascending order.

3. Arrays.copyOf(array, length): copies an array to new array of specified length

4. Arrays.fill(array, value): fills the array with the specified value.

```
int [] numbers = new int[5];
Array.fill(numbers, 10);
S.O.P(Arrays.toString(numbers));
// O/P: [10,10,10,10,10]
```

Types of Array in Java:

1) Program to implement Single Dimensional Array:

O/P:

Python Java C++ C PhP

```
public class Demo{
    public static void main(String []args){
String [] strArray = { "Python", "Java", "C++",
                       "C", "PhP"};
  for ( String i:strArray ){
    S.o.P ( i+" "); }
  S.o.P(" length of Array :" + strArray.length);
  }
}
```

2) Multi-dimensional Array:

Program to show multi-dimensional array with a famous matrix chain multiplication

```
public class Demo {
    public static void main (String []args){
  // Declaring & initializing arrays
```

```
int arr1[][] = {{1,2,3}, {4,5,6}, {7,8,9}};
int arr2[][] = {{2,2,2}, {2,2,2}, {2,2,2}};
// Printing arrays in matrix format
S.O.P("Array-1");
for(int i=0; i<3; i++){ // for-loop for row
    for (int j=0; j<3; j++) { // for-loop for
                              //columns
        S.O.P(arr1[i][j]+" ");
    }
    S.O.P("\n");
}
// Creating an array with same dimensions
int[][] newArr = new int[3][3];
// multiplying arr1 and arr2
for(int i=0; i<3; i++){
    for(int j=0; j<3; j++){
        for(int k=0; k<3; k++){
            newArr[i][j] = 0;
            newArr[i][j] = newArr[i][j] +
                arr1[i][k]*arr2[k][j];
        }
        S.O.P(newArr[i][j]+" ");
    }
    S.O.P();
}
```

Array of objects:

As the name suggest an array of objects is nothing but a list of objects stored in the array. Notice that it does not store objects in an array but stores the reference variable of that objects.

ex: Student Studentobj [] = new Student[3];

```
class Student{
        Student (int id, String name){
    so.pl("student ID"+id+"and name is"+
                "name"); }
    public static void main(String[] args){
    // declaring an array of objects
      Student obj[] = new Student[3];
       obj[0] = new Student (1, "Bharath");
       obj[1] = new Student (4, "Vikas");
       obj[2] = new Student (7, "Shiva");
    }
```

Passing Arrays to Methods:

As we pass variables as a parameter and can also pass the array to method, we donot require adding square brackets with an array name, but we do require that in formal parameters.

Passing an array while calling methods is not required to mention the size of array. If you do so, then it will throw you compile time error.

```
public class Test{
// define function to find max number for
// an array
public static int fmax (int [] arr){
    int max= arr[0];
    for (int i=1; i<arr.length; i++){
        if (max<arr[i]){
            max = arr[i];
        }
    }
    return max;
}
}
```

```
public static void main (String []args){
    int myArray[] = {45, 46, 1, 5, 90, 20};
    // call function name by passing array to it
    int max1 = fmax(myArray);
    S.O.P(max1);
}
```