**3.1 One Dimensional Array Example in Java Using Standard Method**

**public** **class** **OneDimensionalArray** {

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

// Declares an One Dimensional Array of integers

**int** [] anArray;

// Allocates memory for 3 integers

anArray = **new** **int**[**3**];

// initializes first element

anArray[**0**] = **8**;

// initializes second element

anArray[**1**] = **4**;

// initializes third element

anArray[**2**] = **89**;

// Printing the One Dimensional Array

System.out.println("One dimensional array elements are :");

System.out.println("Element at index 0: "+ anArray[**0**]);

System.out.println("Element at index 1: "+ anArray[**1**]);

System.out.println("Element at index 2: "+ anArray[**2**]);

}

}

**Output:**  
**One dimensional array elements are :  
Element at index 0: 8  
Element at index 1: 4  
Element at index 2: 89**

A screenshot of a computer

Description automatically generated

**3.2 One Dimensional Array Example in Java Using Scanner**

**import** **java.util.Scanner**;

**public** **class** **OneDimensionalArrayScanner** {

**public** **static** **void** **main**(String args[])

{

// Creating Scanner Object

Scanner scan = **new** Scanner(System.in);

System.out.println("Enter length of Array: ");

**int** arrLength = scan.nextInt();

**int** [] anArray= **new** **int**[arrLength];

System.out.println("Enter the elements of the Array");

**for**(**int** i = **0**; i < arrLength; i++)

{

anArray[i] = scan.nextInt();

}

// Printing the One Dimensional Array

System.out.println("Displaying One dimensional array elements:");

**for**( **int** i=**0**; i < arrLength ; i++)

{

System.out.print(anArray[i] + " ");

}

}

}

**Output:**  
**Enter length of Array: 4  
Enter the elements of the Array  
8  
4  
89  
2  
Displaying One dimensional array elements:  
8 4 89 2**

Graphical user interface, text, application

Description automatically generated

**3.3 One Dimensional Array Example in Java Using String**

**public** **class** **OneDimensionalArrayString** {

**public** **static** **void** **main**(String args[])

{

// Declaring and Initializing the String Array

String[] strArray = {"Alive is Awesome", "Be in Present","Be Yourself"};

System.out.println("The length of String Array is: "+strArray.length);

// Printing the One Dimensional String Array

System.out.println("Displaying One dimensional String array elements:");

**for**( **int** i=**0**; i < strArray.length ; i++)

{

System.out.println(strArray[i] + " ");

}

}

}

**Output:**  
**The length of String Array is: 3  
Displaying One dimensional String array elements:  
Alive is Awesome  
Be in Present  
Be Yourself**

A screenshot of a computer

Description automatically generated

**Two dimensional array**

|  |
| --- |
| **class** GFG {  **public** **static** **void** main(String[] args)      {    **int**[][] arr = { { 1, 2 }, { 3, 4 } };    **for** (**int** i = 0; i < 2; i++) {  **for** (**int** j = 0; j < 2; j++) {                  System.out.print(arr[i][j] + " ");              }                System.out.println();          }      }  } |

**Output:**

1 2

3 4

A screenshot of a computer

Description automatically generated

|  |
| --- |
| **class** GFG {  **public** **static** **void** main(String[] args)      {    **int**[][][] arr = { { { 1, 2 }, { 3, 4 } },                            { { 5, 6 }, { 7, 8 } } };    **for** (**int** i = 0; i < 2; i++) {    **for** (**int** j = 0; j < 2; j++) {    **for** (**int** k = 0; k < 2; k++) {                        System.out.print(arr[i][j][k] + " ");                  }                    System.out.println();              }              System.out.println();          }      }  } |

**Output:**

1 2

3 4

5 6

7 8

Text, application

Description automatically generated

**Inserting multiple dimensions arrays during runtime**

|  |
| --- |
| **import** java.util.Scanner;    **public** **class** GFGTestCase {  **public** **static** **void** main(          String[] args)      {          // Scanner class to take          // values from console          Scanner scanner = **new** Scanner(System.in);            // totalTestCases = total          // number of TestCases          // eachTestCaseValues =          // values in each TestCase as          // an Array values  **int** totalTestCases, eachTestCaseValues;            // takes total number of          // TestCases as integer number          totalTestCases = scanner.nextInt();            // An array is formed as row          // values for total testCases  **int**[][] arrayMain = **new** **int**[totalTestCases][];            // for loop to take input of          // values in each TestCase  **for** (**int** i = 0; i < arrayMain.length; i++) {              eachTestCaseValues = scanner.nextInt();              arrayMain[i] = **new** **int**[eachTestCaseValues];  **for** (**int** j = 0; j < arrayMain[i].length; j++) {                  arrayMain[i][j] = scanner.nextInt();              }          } // All input entry is done.            // Start executing output          // according to condition provided  **for** (**int** i = 0; i < arrayMain.length; i++) {                // Initialize total number of              // even & odd numbers to zero  **int** nEvenNumbers = 0, nOddNumbers = 0;                // prints TestCase number with              // total number of its arguments              System.out.println(                  "TestCase " + i + " with "                  + arrayMain[i].length + " values:");  **for** (**int** j = 0; j < arrayMain[i].length; j++) {                  System.out.print(arrayMain[i][j] + " ");                    // even & odd counter updated as                  // eligible number is found  **if** (arrayMain[i][j] % 2 == 0) {                      nEvenNumbers++;                  }  **else** {                      nOddNumbers++;                  }              }              System.out.println();                // Prints total numbers of              // even & odd              System.out.println(                  "Total Even numbers: " + nEvenNumbers                  + ", Total Odd numbers: " + nOddNumbers);          }      }  }  // This code is contributed by Udayan Kamble. |

**Input:**

2

2

1 2

3

1 2 3

**Output:**

TestCase 0 with 2 values:

1 2

Total Even numbers: 1, Total Odd numbers: 1

TestCase 1 with 3 values:

1 2 3

Total Even numbers: 1, Total Odd numbers: 2

**Input:**

3

8

1 2 3 4 5 11 55 66

5

100 101 55 35 108

6

3 80 11 2 1 5

**Output:**

TestCase 0 with 8 values:

1 2 3 4 5 11 55 66

Total Even numbers: 3, Total Odd numbers: 5

TestCase 1 with 5 values:

100 101 55 35 108

Total Even numbers: 2, Total Odd numbers: 3

TestCase 2 with 6 values:

3 80 11 2 1 5

Total Even numbers: 2, Total Odd numbers: 4

Graphical user interface, text, application

Description automatically generated