

Java Programming Assignment– 8

PART A-Theory Questions

1) How are Strings stored in memory?

Answer :

String :- String is a class in Java (java.lang.String) used to create and manipulate a sequence of characters, where once a String object is created, its value cannot be changed.

- String is a **non-primitive data type**
- It is a **final class**
- Strings are **immutable**
- Stored in **String Constant Pool or Heap**
- Supports many built-in methods (like length(), charAt(), equals())

How Strings are stored in memory (Java)

In Java, **Strings are stored in two main memory areas:**

1. **String Constant Pool (SCP)** – part of Heap
2. **Heap memory (normal heap objects)**

1. String Constant Pool (SCP)

- SCP is a **special area inside the Heap**
- It stores **unique string literals**
- Java reuses the same object to save memory

e.g.

```
String s1 = "Java";  
String s2 = "Java";  
System.out.println(s1 == s2);  
System.out.println(s1.equals(s2));
```

What happens in memory?

- "Java" is created **only once** in SCP
- s1 and s2 both **refer to the same object**

2. Heap Memory (using new keyword)

When you create a string using new, **a new object is always created**, even if the content is the same.

e.g. :

```
String s1 = new String("java");
String s2 = new String("Java");
System.out.println(s1 == s2);
System.out.println(s1.equals(s2));
```

Memory behavior:

- "Java" → stored in SCP (if not already present)
- new String("Java") → stored as **separate object in Heap**

2) What is an array in Java?

Answer :

An array in Java is an object that stores multiple values of the same data type in a continuous memory location and is accessed using an index.

Characteristics of Arrays in Java

- Array is an **object**
- Stored in **Heap memory**
- Size is **fixed**
- Stores **homogeneous data** (same type)
- Index starts from **0**
- Can store **primitive** and **non-primitive** data
- Supports **random access**

Syntax : **dataType[] arrayName;**

3) How do you declare and initialize an array?

Answer :

e.g. :

a) Declaration and Static initialization :

```
int[] a = {10, 20, 30, 40};
```

```
int a[] = {10, 20, 30, 40};
```

```
String[] names ;
```

b) Dynamic initialization :

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

```
a[0] = 10;
```

```
a[1] = 20;
```

c) Array creation (Instanciation) :

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

4) What are jagged arrays?

Answer :

Definition :

A jagged array in Java is a multi-dimensional array in which each row can have a different number of columns.

Types : two types.

- 1) Irregular Array.
- 2) Ragged Array.

Why it called jagged array :- In a normal 2D array, all rows have the same length, but in jagged array rows have unequal array.

Characteristics :-

- It is a **2D array with variable column size**
- Stored in **Heap memory**
- Each row is a **separate 1D array**
- Memory efficient when rows need different sizes
- Index starts from **0**
- Size of columns can be decided **dynamically**

Declaration of jagged array :-

```
int [][] a; or int a[][];
```

Creation of jagged array :

1) Create row reference:

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

created 3 rows. Column size is not defined.

2) Define column size for each row :

```
a[0] = new int[2];
```

```
a[1] = new int[4];
```

```
a[2] = new int[1];
```

note : 0,1,2 are rows. And 2,4,1 are row size respectively.

3) Initialization :

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

```
a[0] = new int[]{1, 2};
```

```
a[1] = new int[]{3, 4, 5, 6};
```

```
a[2] = new int[]{7};
```

5) What is an ArrayIndexOutOfBoundsException?

Answer :

ArrayIndexOutOfBoundsException is a runtime exception that occurs when a program tries to access an array element using an index that is less than 0 or greater than or equal to the array size. It happens because array has fixed index size.

PART B-Programming Questions

1) Write a program to count even and odd numbers in an array.

Answer :

```
import java.util.Scanner;  
  
public class CountEvenOdd {  
  
    public static void main(String[] args) {
```

```

// TODO Auto-generated method stub

Scanner sc = new Scanner(System.in);

System.out.println("Enter size of array : ");

int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter Elements of array : ");

for (int i = 0; i < n; i++) {

    arr[i] = sc.nextInt();

}

// counting even and odd elements in array.

int countEven = 0;

int countOdd = 0;

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

    if (arr[i] % 2 == 0) {

        countEven++;

    } else {

        countOdd++;

    }

}

System.out.println("Count of even numbers : "+countEven);

System.out.println("Count of odd numbers : "+countOdd);

}

}

```

Q. 2) Write a program to copy all elements of one array to another.

Answer :

```
import java.util.Scanner;
```

```

public class CopyArray {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter array size : ");
        int n = sc.nextInt();
        int[] arr1 = new int[n];
        System.out.println("Enter element of array : ");
        for (int i = 0; i < n; i++) {
            arr1[i] = sc.nextInt();
        }
        int[] arr2 = new int[n];
        // copying arr1 element in arr2 array
        for (int i = 0; i < n; i++) {
            arr2[i] = arr1[i];
        }
        // print second array
        for (int i = 0; i < arr2.length; i++) {
            System.out.print(arr2[i] + " ");
        }
    }
}

```

Q. 3) Write a program to search an element in an array.

Answer :

```

import java.util.Scanner;
public class SearchArrayElement {
    public static void main(String[] args) {

```

```

// TODO Auto-generated method stub

Scanner sc = new Scanner(System.in);

System.out.println("Enter size of array : ");

int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter array element : ");

for (int i = 0; i < n; i++) {

    arr[i] = sc.nextInt();

}

System.out.println("Enter element to search : ");

int elt = sc.nextInt();

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

    if (arr[i] == elt) {

        System.out.println("Element found at index "+i);

    }

}

}

```

Q. 4) Write a program to merge two arrays.

Answer :

```

import java.util.Scanner;

public class MergeArray {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a size : ");

        int n1 = sc.nextInt();

```

```

int[] arr1 = new int[n1];
System.out.println("Enter element : ");
for (int i = 0; i < n1; i++) {
    arr1[i] = sc.nextInt();
}

System.out.println("Element size : ");
int n2 = sc.nextInt();
int[] arr2 = new int[n2];
System.out.println("Enter Element : ");
for (int i = 0; i < n2; i++) {
    arr2[i] = sc.nextInt();
}

int arr3[] = new int[n1+n2];
int index = 0;
for (int i = 0; i < n1; i++) {
    arr3[index++] = arr1[i];
}
for (int i = 0; i < n2; i++) {
    arr3[index++] = arr2[i];
}
for (int x : arr3) {
    System.out.print(x+" ");
}
}
}

```

Q. 5) Write a program to remove duplicate elements from an array.

Answer :

```
import java.util.Scanner;

public class RemoveDuplicate {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a size : ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter Element : ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        for (int i = 0; i < n; i++) {
            for (int j = i+1; j < n; j++) {
                if (arr[i] == arr[j]) {
                    arr[j] = arr[j+1];
                    n--;
                }
            }
        }
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i]+ " ");
        }
    }
}
```

