

Java Array Programs with Concepts and Solutions

Array Concepts in Java

An array in Java is a data structure that allows you to store multiple values of the same type in a single variable.

Key points:

- Fixed size: Once declared, the size of the array cannot be changed.
- Index-based: Index starts from 0.
- Can store primitives or objects.
- Useful for storing large amounts of data efficiently.

Declaration: `int[] arr;`

Initialization: `arr = new int[5];`

Combined: `int[] arr = {10, 20, 30, 40, 50};`

Access: `arr[0]`, `arr.length`

1. Print All Elements of an Array

Description: This program prints each element of an integer array using a for loop.

```
public class PrintArray {
    public static void main(String[] args) {
        int[] arr = {10, 20, 30, 40};

        for (int i = 0; i < arr.length; i++) {
            System.out.println("Element at index " + i + " is " + arr[i]);
        }
    }
}
```

2. Find the Sum of All Array Elements

Description: This program calculates the sum of all elements in the array.

```
public class ArraySum {
    public static void main(String[] args) {
        int[] numbers = {5, 10, 15, 20};
        int sum = 0;

        for (int n : numbers) {
            sum += n;
        }

        System.out.println("Sum = " + sum);
    }
}
```

3. Find the Largest Element in an Array

Description: This program finds the maximum value in the array.

```
public class MaxInArray {
    public static void main(String[] args) {
        int[] nums = {8, 23, 45, 12, 67};
        int max = nums[0];
    }
}
```

```

        for (int n : nums) {
            if (n > max) {
                max = n;
            }
        }

        System.out.println("Maximum: " + max);
    }
}

```

4. Find the Smallest Element in an Array

Description: This program finds the minimum value in the array.

```

public class MinInArray {
    public static void main(String[] args) {
        int[] nums = {25, 12, 5, 78, 1};
        int min = nums[0];

        for (int n : nums) {
            if (n < min) {
                min = n;
            }
        }

        System.out.println("Minimum: " + min);
    }
}

```

5. Count Even and Odd Numbers

Description: This program counts how many elements are even and how many are odd.

```

public class CountEvenOdd {
    public static void main(String[] args) {
        int[] nums = {10, 21, 4, 15, 8};
        int even = 0, odd = 0;

        for (int n : nums) {
            if (n % 2 == 0) even++;
            else odd++;
        }

        System.out.println("Even: " + even);
        System.out.println("Odd: " + odd);
    }
}

```

6. Reverse an Array

Description: This program prints the array in reverse order.

```

public class ReverseArray {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 5};

        for (int i = arr.length - 1; i >= 0; i--) {
            System.out.print(arr[i] + " ");
        }
    }
}

```

7. Copy One Array into Another

Description: This program copies elements from one array to another.

```
public class CopyArray {
    public static void main(String[] args) {
        int[] original = {5, 10, 15};
        int[] copy = new int[original.length];

        for (int i = 0; i < original.length; i++) {
            copy[i] = original[i];
        }

        System.out.println("Copied Array:");
        for (int num : copy) {
            System.out.print(num + " ");
        }
    }
}
```

8. Calculate Average of Array Elements

Description: This program finds the average of all elements in the array.

```
public class AverageArray {
    public static void main(String[] args) {
        int[] marks = {70, 80, 90, 100};
        int total = 0;

        for (int mark : marks) {
            total += mark;
        }

        double avg = (double) total / marks.length;
        System.out.println("Average = " + avg);
    }
}
```

9. Search an Element in Array

Description: This program checks whether a given element exists in the array.

```
public class SearchElement {
    public static void main(String[] args) {
        int[] arr = {11, 22, 33, 44, 55};
        int key = 33;
        boolean found = false;

        for (int num : arr) {
            if (num == key) {
                found = true;
                break;
            }
        }

        if (found)
            System.out.println("Element Found!");
        else
            System.out.println("Element Not Found!");
    }
}
```

10. Sort an Array (Ascending Order)

Description: This program sorts the array in ascending order using Arrays.sort().

```
import java.util.Arrays;
public class SortArray {
    public static void main(String[] args) {
        int[] arr = {40, 10, 30, 20};

        Arrays.sort(arr);

        System.out.println("Sorted Array:");
        for (int num : arr) {
            System.out.print(num + " ");
        }
    }
}
```