Java Array Practice Exercises

Foundational Exercises: The Basics

These exercises will help you practice the fundamental concepts of creating, initializing, and accessing arrays.

Exercise 1: Create and Print an Array

1. Declare an array of integers named `numbers` that can hold 5 elements.
2. Initialize this array with the values: 10, 20, 30, 40, 50.
3. Use a `for` loop and the `.length` property to print each element of the array to the console.

Exercise 2: Boolean Flags

1. [cite_start]As your notes show, you can create arrays of any type. [cite_start]Declare a boolean array named `status` of size 4.
2. Initialize the first element to `true` and the rest to `false`.
3. [cite_start]Print the value of the 3rd element (at index 2).

Exercise 3: User-Defined Array

[cite_start]Write a program that does the following, similar to the example in your notes:

1. Asks the user to enter a size for an array.
2. Creates an integer array of the specified size.
3. Uses a `for` loop to ask the user to enter a value for each index of the array.
4. Prints all the elements entered by the user.

Intermediate Exercises: Array Operations

These exercises involve performing calculations and manipulations on array data.

Exercise 4: Sum and Average

1. Create an integer array and initialize it with at least 6 numbers.

2. Write code to calculate the sum of all elements in the array.
3. Calculate the average of the elements and print both the sum and the average.

Exercise 5: Find the Maximum and Minimum

1. Declare an integer array with a set of positive and negative numbers.
2. Write the logic to find the largest and smallest numbers in the array.
3. Print the maximum and minimum values you found.

Exercise 6: Search for an Element

1. Create a `String` array initialized with a list of five names.
2. Ask the user to enter a name to search for.
3. Write a loop to check if the entered name exists in the array.
4. If the name is found, print its index. If not, print a "Name not found" message.

Advanced Exercises: Arrays and Methods

[cite_start]These exercises focus on the concepts of passing and returning arrays from methods, as shown in your notes.

Exercise 7: Method to Display an Array

1. In your `main` method, create an integer array.
2. [cite_start]Write a separate method called `printArray` that accepts an integer array as an argument (e.g., `void printArray(int[] arr)`).
3. This method should loop through the array and print each element.
4. Call this method from `main` and pass your array to it.

Exercise 8: Method to Create an Array

1. Write a method called `createRandomArray` that takes an integer `size` as input.
2. Inside this method, create an array of the given `size`.
3. Fill the array with random numbers (Hint: use the `Math.random()` function).
4. [cite_start]The method should return the newly created array, similar to the `input()` method in your notes.
5. In your `main` method, call `createRandomArray` to get an array and then print its elements.

Exercise 9: Reverse an Array

1. Write a method that takes an integer array as an argument.
2. This method should create a *new* array that contains all the elements of the original array but in reverse order.
3. Return the new, reversed array.
4. In `main`, create an array, pass it to your reverse method, and then print the elements of the returned (reversed) array.

Conceptual Question

Exercise 10: Predict the Error

Look at the following code snippet. Without running it, what will happen when this code is executed? [cite_start]Explain your answer based on your notes.

```
public class TestException {
    public static void main(String[] args) {
        int[] scores = new int[5]; // Array has indices 0, 1, 2, 3, 4
        scores[0] = 95;
        scores[1] = 87;
        scores[2] = 91;
        scores[3] = 88;
        scores[4] = 99;

        // This loop tries to access one element too far
        for (int i = 0; i <= scores.length; i++) {
            System.out.println(scores[i]);
        }
    }
}
```