

# Java Coding Challenges Interview Questions

Topics : [Java Interview Questions](#)

Written on [November 18, 2023](#)

## Coding Challenge 1: Palindrome Check

Write a Java program to check if a given string is a palindrome.

**Solution:**

```
import java.util.Scanner;

public class PalindromeCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        if (isPalindrome(input)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
    }

    private static boolean isPalindrome(String str) {
        String reversed = new StringBuilder(str).reverse().toString();
        return str.equals(reversed);
    }
}
```

## Coding Challenge 2: Fibonacci Series

**Problem:** Write a Java program to print the Fibonacci series up to a given number of terms.

**Solution:**

```
import java.util.Scanner;

public class FibonacciSeries {
    public static void main(String[] args) {
```

```

Scanner scanner = new Scanner(System.in);
System.out.print("Enter the number of terms in the Fibonacci series:
");
int n = scanner.nextInt();

printFibonacciSeries(n);
}

private static void printFibonacciSeries(int n) {
    int a = 0, b = 1;

    System.out.println("Fibonacci Series:");

    for (int i = 0; i < n; i++) {
        System.out.print(a + " ");

        int temp = a;
        a = b;
        b = temp + b;
    }
}
}

```

### Coding Challenge 3: Prime Number Check

**Problem:** Write a Java program to check if a given number is prime.

**Solution:**

```

import java.util.Scanner;

public class PrimeNumberCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        if (isPrime(number)) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }
    }

    private static boolean isPrime(int num) {
        if (num <= 1) {
            return false;
        }

        for (int i = 2; i <= Math.sqrt(num); i++) {

```

```

        if (num % i == 0) {
            return false;
        }
    }
    return true;
}
}

```

## Reverse a String

**Problem:** Write a Java program to reverse a given string without using any built-in functions or libraries.

**Solution:**

```

public class ReverseString {
    public static void main(String[] args) {
        String original = "Hello, World!";
        String reversed = reverseString(original);
        System.out.println("Original: " + original);
        System.out.println("Reversed: " + reversed);
    }

    private static String reverseString(String str) {
        char[] chars = str.toCharArray();
        int left = 0;
        int right = chars.length - 1;

        while (left < right) {
            char temp = chars[left];
            chars[left] = chars[right];
            chars[right] = temp;
            left++;
            right--;
        }

        return new String(chars);
    }
}

```

## Find the First Non-Repeated Character

**Problem:** Write a Java program to find the first non-repeated character in a string.

**Solution:**

```

public class FirstNonRepeatedChar {
    public static void main(String[] args) {
        String input = "programming";
    }
}

```

```

char result = findFirstNonRepeatedChar(input);
System.out.println("First non-repeated character: " + result);
}

private static char findFirstNonRepeatedChar(String str) {
    int[] charCount = new int[256];

    for (char ch : str.toCharArray()) {
        charCount[ch]++;
    }

    for (char ch : str.toCharArray()) {
        if (charCount[ch] == 1) {
            return ch;
        }
    }

    return '\0'; // Return null character if no non-repeated character is found
}
}

```

You can try as below for practice.

### **1. Reverse a String**

Write a Java program to reverse a given string without using any built-in functions or libraries.

### **2. Check for Palindrome**

Write a Java program to check if a given string is a palindrome.

### **3. Find the First Non-Repeated Character**

Write a Java program to find the first non-repeated character in a string.

### **4. Fibonacci Series**

Write a Java program to generate the Fibonacci series up to a specified number of terms.

### **5. Factorial Calculation**

Write a Java program to calculate the factorial of a given number.

### **6. Check for Prime Number**

Write a Java program to check if a given number is prime.

### **7. Find the Missing Number**

Write a Java program to find the missing number in an array of 1 to N.

## **8. Rotate an Array**

Write a Java program to rotate elements in an array to the right by a specified number of positions.

## **9. Check for Anagrams**

Write a Java program to check if two strings are anagrams of each other.

## **10. Implement a Stack**

Write a Java program to implement a stack with push, pop, and peek operations.

## **11. Implement a Queue using Stacks**

Write a Java program to implement a queue using two stacks.

## **12. Find the Maximum Subarray Sum**

Write a Java program to find the maximum sum of a subarray in an array of integers.

## **13. Reverse a Linked List**

Write a Java program to reverse a linked list.

## **14. Binary Search**

Write a Java program to perform a binary search on a sorted array.

## **15. Implement a Binary Tree**

Write a Java program to implement a binary tree with basic operations like insertion and traversal.

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)