














# **Top Java Number Programs** **Every QA Engineer Should** **Master Before Their Next** **Interview!** 🎯💻



**Swipe>>**

@ajaytripathi

1.  **Check if a Number is Prime**
2.  **Generate Fibonacci Series**
3.  **Swap Two Numbers Without Using a Temporary Variable**
4.  **Count the Number of Digits in an Integer**
5.  **Find Duplicate Numbers in a List**
6.  **Find Unique (Non-Duplicate) Numbers in a List**
7.  **Calculate the Factorial of a Number**
8.  **Reverse a Number**
9.  **Find the Smallest Number in a List**
10.  **Find the Largest Number in a List**
11.  **Calculate the Sum of Digits of a Number**
12.  **Check if a Number is an Armstrong Number**
13.  **Check if a Number is a Palindrome**

**Swipe>>**

## 1 Check if a Number is Prime

PrimeCheck.java

```
public class PrimeCheck {  
    public static void main(String[] args) {  
        int num = 29;  
        boolean isPrime = true;  
  
        if (num <= 1) {  
            isPrime = false;  
        } else {  
            for (int i = 2; i <= Math.sqrt(num); i++) {  
                if (num % i == 0) {  
                    isPrime = false;  
                    break;  
                }  
            }  
        }  
  
        if (isPrime)  
            System.out.println(num + " is a Prime Number.");  
        else  
            System.out.println(num + " is NOT a Prime Number.");  
    }  
}
```

Swipe>>

## 2 *Generate Fibonacci Series*

FibonacciSeries.java

```
public class FibonacciSeries {  
    public static void main(String[] args) {  
        int count = 10;  
        int first = 0, second = 1;  
  
        System.out.print("Fibonacci Series up to " + count + ": ");  
  
        for (int i = 1; i <= count; i++) {  
            System.out.print(first + " ");  
            int next = first + second;  
            first = second;  
            second = next;  
        }  
    }  
}
```

Swipe>>

### 3 *Swap Two Numbers Without Using a Temporary Variable*

SwapWithoutTemp.java

```
public class SwapWithoutTemp {  
    public static void main(String[] args) {  
        int a = 5, b = 10;  
  
        System.out.println("Before Swap: a = " + a + ", b = " + b);  
  
        a = a + b;  
        b = a - b;  
        a = a - b;  
  
        System.out.println("After Swap: a = " + a + ", b = " + b);  
    }  
}
```

Swipe>>

## 4 *Count the Number of Digits in an Integer*

CountDigits.java

```
public class CountDigits {  
    public static void main(String[] args) {  
        int num = 123456;  
        int count = 0;  
  
        while (num != 0) {  
            num /= 10;  
            count++;  
        }  
  
        System.out.println("Number of digits: " + count);  
    }  
}
```

Swipe>>



## 5 Find Duplicate Numbers in a List

FindDuplicates.java

```
public class FindDuplicates {  
    public static void main(String[] args) {  
        List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 2, 5, 1, 6);  
        Set<Integer> seen = new HashSet<>();  
        Set<Integer> duplicates = new HashSet<>();  
  
        for (int num : numbers) {  
            if (!seen.add(num)) {  
                duplicates.add(num);  
            }  
        }  
  
        System.out.println("Duplicate Numbers: " + duplicates);  
    }  
}
```

Swipe>>

## 6 Find Unique (Non-Duplicate) Numbers in a List

FindUniques.java

```
public class FindUniques {  
    public static void main(String[] args) {  
        List<Integer> numbers = Arrays.asList(1, 2, 2, 3, 4, 4, 5);  
        Map<Integer, Integer> countMap = new HashMap<>();  
  
        for (int num : numbers) {  
            countMap.put(num, countMap.getOrDefault(num, 0) + 1);  
        }  
  
        System.out.print("Unique Numbers: ");  
        for (int num : countMap.keySet()) {  
            if (countMap.get(num) == 1) {  
                System.out.print(num + " ");  
            }  
        }  
    }  
}
```

Swipe>>



## 7 Calculate the Factorial of a Number

Factorial.java

```
public class Factorial {  
    public static void main(String[] args) {  
        int num = 5;  
        long factorial = 1;  
  
        for (int i = 1; i <= num; i++) {  
            factorial *= i;  
        }  
  
        System.out.println("Factorial of " + num + " is " + factorial);  
    }  
}
```

Swipe>>

## 8 *Reverse a Number*

ReverseNumber.java

```
public class ReverseNumber {  
    public static void main(String[] args) {  
        int num = 12345;  
        int reversed = 0;  
  
        while (num != 0) {  
            int digit = num % 10;  
            reversed = reversed * 10 + digit;  
            num /= 10;  
        }  
  
        System.out.println("Reversed Number: " + reversed);  
    }  
}
```

Swipe>>

## 9 Find the Smallest Number in a List

SmallestInList.java

```
import java.util.*;

public class SmallestInList {
    public static void main(String[] args) {
        List<Integer> list = Arrays.asList(10, 4, 2, 99, 1);
        int min = Collections.min(list);

        System.out.println("Smallest Number: " + min);
    }
}
```

Swipe>>

## 10 Find the Largest Number in a List

LargestInList.java

```
import java.util.*;

public class LargestInList {
    public static void main(String[] args) {
        List<Integer> list = Arrays.asList(10, 4, 2, 99, 1);
        int max = Collections.max(list);

        System.out.println("Largest Number: " + max);
    }
}
```

Swipe>>

## 1 1 Calculate the Sum of Digits of a Number

ArmstrongNumber.java

```
public class SumOfDigits {  
    public static void main(String[] args) {  
        int num = 1234;  
        int sum = 0;  
  
        while (num != 0) {  
            sum += num % 10;  
            num /= 10;  
        }  
  
        System.out.println("Sum of Digits: " + sum);  
    }  
}
```

Swipe>>

## 1 2 Check if a Number is an Armstrong Number

ArmstrongNumber.java

```
public class ArmstrongNumber {
    public static void main(String[] args) {
        int num = 153, original = num, sum = 0;

        while (num != 0) {
            int digit = num % 10;
            sum += Math.pow(digit, 3);
            num /= 10;
        }

        if (sum == original)
            System.out.println(original + " is an Armstrong Number.");
        else
            System.out.println(original + " is NOT an Armstrong Number.");
    }
}
```

Swipe>>



## 13 Check if a Number is a Palindrome

PalindromeNumber.java

```
public class PalindromeNumber {  
    public static void main(String[] args) {  
        int num = 121, original = num, reversed = 0;  
  
        while (num != 0) {  
            int digit = num % 10;  
            reversed = reversed * 10 + digit;  
            num /= 10;  
        }  
  
        if (original == reversed)  
            System.out.println(original + " is a Palindrome.");  
        else  
            System.out.println(original + " is NOT a Palindrome.");  
    }  
}
```

Swipe>>