Java Practice Programs

This Java exercise is designed to deepen your understanding and refine your Java coding skills, these programs offer hands-on experience in solving real-world problems, reinforcing key concepts, and mastering Java programming fundamentals

Instructions

Each solution should have a program file e.g question1.java
All programs should use one main function.
All answers to solution should act as a project.
No use of formula in solving problems

1. Write Hello World Program in Java

2. Write a Program in Java to Add two Numbers.

```
Input: 2 3
Output: 5
```

3. Write a Program to Swap Two Numbers

```
Input: a=2  b=5
Output: a=5  b=2
```

4. Write a Java Program to convert Integer numbers and Binary numbers.

```
Input: 9
Output: 1001
```

5. Write a Program to Find Factorial of a Number in Java.

```
Input: 5
Output: 120
```

6. Write a Java Program to Add two Complex Numbers.

```
Input:   1+2i
           4+5i
Output: 5+7i
```

7. Write a Program to Calculate Simple Interest in Java

```
Input :  P = 10000  R = 5  T = 5
Output : 2500
```

8. Write a Program to Print the Pascal’s Triangle in Java

```
Input : N = 5
Output:
        1
      1   1
    1   2   1
  1   3   3   1
```

```
  1   4   6   4   1
1   5   10   10   5   1
```

9. Write a Program to Find Sum of Fibonacci Series Number

```
Input: n = 4
Output: 33

Explaination: Sum of numbers at even indexes = 0 + 1 + 3 + 8 + 21 = 33.
```

10. Write a Program to Print Pyramid Number Pattern in Java.

```
    *
   ***
  *****
 *******
```

11. Write a Java Program to Print Pattern.

```
* * * * * *
*         *
*         *
*         *
*         *
* * * * * *
```

12. Write a Java Program to Print Pattern.

```
         1
       2 1 2
     3 2 1 2 3
   4 3 2 1 2 3 4
  5 4 3 2 1 2 3 4 5
6 5 4 3 2 1 2 3 4 5 6
```

13. Java Program to Print Patterns.

```
    *
   ***
  *****
 *******
*********
***********
 *********
  *******
   *****
    ***
     *
```

14. Write a Java Program to Compute the Sum of Array Elements.

```
Input: [ 2, 4, 6, 7, 9]
Output: 28
```

15. Write a Java Program to Find the Largest Element in Array

```
Input: [ 7, 2, 5, 1, 4]
Output: 7
```

16. Write Java Program to Find the Tranpose of Matrix

```
Input:
[ [ 1, 2, 3 ]
  [ 4, 5, 6 ]
  [ 7, 8, 9 ] ]
Output:
[ [ 1, 4, 7]
  [ 2, 5, 8]
  [ 3, 6, 9] ]
```

17. Java Array Program For Array Rotation

```
Input: arr[] = {1, 2, 3, 4, 5, 6, 7},  d = 2
Output:  3 4 5 6 7 1 2

Explaination: d=2 so 2 elements are rotated to the end of the array.
So, 1 2 is rotated back
So, Final result: 3, 4, 5, 6, 7, 1, 2
```

18. Java Array Program to Remove Duplicate Elements From an Array

```
Input: [ 1, 2, 2, 3, 3, 3, 4, 5 ]
Output: [ 1, 2, 3, 4, 5 ]
```

19. Java Array Program to Remove All Occurrences of an Element in an Array

```
Input: array = [ 1, 2, 1, 3, 5, 1 ] , key = 1
Output: [2, 3, 5]

Explaination: all the occurrences of element 1 is removed from the
array So, array becomes from
[ 1, 2, 1, 3, 5, 1 ]  to
Final result: [2, 3, 5]
```

20. Java program to check whether a string is a Palindrome

```
Input: "racecar"
Output: Yes

Explaination: As racerar after reversing becomes racecar. As Reverse of String is same as String so it is Palindrome
```

21. Java String Program to Check Anagram

```
Input: str1 = "Silent"
          str2 ="Listen"
Output: Strings are Anagram

Explaination: As all the elements in str1 are exact same as to create str2 .
i.e., we can create str2 using elements of str1 without removing any element or removing any extra element.
```

22. Java String Program to Reverse a String

```
Input: str= "Geeks"
Output: "skeeG"
```

23. Java String Program to Remove leading zeros

```
Input : 00000123569
Output : 123569
```

Java Practice Problems for Searching Algorithms

24. Write a Java Program for Linear Search.
    Time Complexity: *O(N)*
    Space Complexity: *O(N)*

25. Write a Binary Search Program in Java.
    Time Complexity: *O(logN)*
    Space Complexity: *O(N)*

Practice Problems in Java Sorting Algorithms

26. Java Program for Bubble Sort.
    Time Complexity: *$O(N_2)$*
    Space Complexity: *O(1)*

27. Write a Program for Insertion Sort in Java.
    Time Complexity: *O(N_2)*
    Space Complexity: *O(1)*

28. Java Program for Selection Sort.
    Time Complexity: *O(N_2)*
    Space Complexity: *O(1)*

29. Java Program for Merge Sort.
    Time Complexity: *O(N logN)*
    Space Complexity: *O(N)*

30. Java Program for QuickSort.
    Time Complexity: *O(N logN)*
    Space Complexity: *O(1)*