Assignment - 4

Object-Oriented Programming in Java

Name: Kamithkar Vinod
Course: PG DAC AUGUST 2025
Form No: 250500480
Date: 14-09-2025

Problem 1: Print Numbers 10 to 110

Task: Write a program to print numbers 10 to 110.

Code:

```java
class _1PrintNumbers {
    public static void main(String[] args) {
        for (int i = 10; i <= 110; i++) {
            System.out.print(i + " ");
        }
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>javac _1PrintNumbers.java

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _1PrintNumbers
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>
```

Problem 2: Sum of All Numbers from 1 to 100

Task: Write a program to calculate the sum of all numbers from 1 to 100.

Code:

```java
class _2SumOfNumbers {
    public static void main(String[] args) {
        int sum = 0;
        for (int i=1; i<=100; i++) {
            sum += i;
        }
        System.out.println("Sum of No's 1 to 100: " + sum);
```

```
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>javac _2SumOfNumbers.java

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _2SumOfNumbers
Sum of No's 1 to 100: 5050
```

Problem 3: Multiplication Table of a Given Number

Task: Write a program to print the multiplication table of a given number.

Code: —

```
import java.util.Scanner;
class _3MultiplicationTable {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the Number to do
           MultiplicationTable: ");
        int n = scanner.nextInt();

        for (int i=1; i<=10; i++) {
            System.out.println(n +" * "+ i + " = " + n*i);
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>javac _3MultiplicationTable.java

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _3MultiplicationTable
Enter the Number to do MultiplicationTable:
9
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
9 * 10 = 90
```

Problem 4: Factorial Given Number

Task: Write a program to find the factorial of a given number.

Code:

```java
import java.util.Scanner;
class _4Factorial {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a Number to find factorial: ");
        int f = scanner.nextInt();
        int total = 1;
        for(int i=0; i<f; i++) {
            total *= (f-i);
        }
        System.out.println("The factorial of "+f+" is "+total);
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>javac _4Factorial.java

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _4Factorial
Enter a Number to find factorial: 5
The factorial of 5 is 120
```

Problem 5: Check Prime Number or Not

Task: Write a program to check if a given number is prime.

Code:

```java
class _5Prime {
    public static void main(String[] args) {
        // generate random number 1 to 1000
        int n = 1 + (int) (Math.random() * 1000);
        int count = 0, i;

        for(i=1; i<=n/2; i++) {
            if (n%i == 0){
                count++;
                if (count > 1)
                    break;
            }
        }
        System.out.println("No of iterations = " + (i-1));
        if (count == 1)
            System.out.println(n + " is a Prime Number");
        else
            System.out.println(n + " is not a Prime Number");
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _5Prime
No of iterations = 1
670 is not a Prime Number

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _5Prime
No of iterations = 15
31 is a Prime Number
```

Problem 6: Fibonacci Series

Task: Write a program to print the Fibonacci series up to a given number of terms.

Code: —

```java
class _6Fibonacci {
    public static void main(String[] args) {
        int n = 1 + (int) (Math.random()*10);
        int first = 0, second = 1;

        System.out.println("Fibonacci Series upto "+n+" terms: ")
           ;

        for(int i = 1; i<=n; i++){
            System.out.print(first + " ");

            int next = first + second;
            first = second;
            second = next;
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>javac _6Fibonacci.java

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _6Fibonacci
Fibonacci Series upto 6 terms:
0 1 1 2 3 5
```

Problem 7: Sum of digits of a Number

Task: Write a program to calculate the sum of digits of a given number.

Code: —

```java
class _7SumOfDigits {
    public static void main(String[] args) {
        int num = 101 + (int) (Math.random() * 900);
        int sum = 0;
        int temp = num;
```

```
6
7           while (temp != 0) {
8               int digit = temp % 10;
9               sum += digit;
10              temp /= 10;
11          }
12          System.out.println("Sum of digits of "+num+" is :" + sum)
                ;
13      }
14  }
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _7SumOfDigits
Sum of digits of 750 is :12

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _7SumOfDigits
Sum of digits of 338 is :14
```

Problem 8: Check Palindrome Number or Not

Task: Write a program to check if a given number is a palindrome.

Code: —

```
1   class _8Palindrome {
2       public static void main(String[] args) {
3
4           int num = 101 + (int) (Math.random() * 900);
5           int original = num;
6           int reversed = 0;
7
8           while (num != 0) {
9               int digit = num % 10;
10              reversed = (reversed * 10) + digit;
11              num /= 10;
12          }
13          if (original == reversed)
14              System.out.println(original + " is a Palindrome");
15          else
16              System.out.println(original + " is not a Palindrome")
                    ;
17      }
18  }
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _8Palindrome
565 is a Palindrome

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _8Palindrome
401 is not a Palindrome
```

Problem 9: Odd Numbers 1 to 50

Task: 9. Write a program to find the sum of all odd numbers between 1 and 50

Code:

```java
class _9SumOfOdd {
    public static void main(String[] args) {
        int sum = 0;
        for(int i = 1; i<= 50; i++) {
            if (i % 2 != 0)
                sum += i;
        }
        System.out.println("Sum of Odd Numbers 1 to 50 is: "+sum)
           ;
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _9SumOfOdd
Sum of Odd Numbers 1 to 50 is: 625

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>
```

Problem 10: Even Numbers 1 to 50

Task: Write a program to find the sum of all even numbers between 1 and 50.

Code:

```java
class _10SumOfEven {
    public static void main(String[] args) {
        int sum = 0;
        for(int i = 1; i<= 50; i++) {
            if (i % 2 == 0)
                sum += i;
        }
        System.out.println("Sum of Odd Numbers 1 to 50 is: "+sum)
           ;
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment>java _10SumOfEven.java
Sum of Odd Numbers 1 to 50 is: 650
```

Problem 11: Check Armstrong Number

Task: Write a program to check if a given number is Armstrong.

Code:

```java
import java.util.Scanner;
class _11Armstrong {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a Number: ");
        int num = scanner.nextInt();

        int sum = 0;
        int n = num;

        while (n != 0){
            int digit = n % 10;
            sum += Math.pow(digit, 3);
            n /= 10;
        }
        if (num == sum)
            System.out.println(num + " is an Armstrong Number");
        else
            System.out.println(num + " is not an Armstrong Number
               ");
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _11Armstrong
Enter a Number: 153
153 is an Armstrong Number

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _11Armstrong
Enter a Number: 143
143 is not an Armstrong Number
```

Problem 12: Reverse Number

Task: Write a program to reverse a given number.

Code:

```java
class _12ReverseNumber {
    public static void main(String[] args) {
        int num = 101 + (int) (Math.random() * 900);
        int reverse = 0;
        int n = num;
        while(n != 0) {
            int digit = n % 10;
            reverse = (reverse * 10) + digit;
            n /= 10;
        }
        System.out.println("The reverse of "+num+" is: "+reverse)
           ;
```

```
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _12ReverseNumber.java
The reverse of 929 is: 929

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _12ReverseNumber.java
The reverse of 280 is: 82
```

Problem 13: Calculate Power of the loop

Task: Write a program to calculate the power of a number using a loop.

Code: —

```java
import java.util.Scanner;
class _13PowerCalc {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a Base Number: ");
        int base = scanner.nextInt();
        System.out.print("Enter a exponent: ");
        int exp = scanner.nextInt();

        long result = 1;

        for(int i=0; i<exp; i++){
            result *= base;
        }
        System.out.println(base+" to the power of "+exp+" = "+
           result);
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _13PowerCalc.java
Enter a Base Number: 25
Enter a exponent: 2
25 to the power of 2 = 625
```

Problem 14: GCD of Two Numbers

Task: Write a program to find the greatest common divisor (GCD) of two numbers.

Code:

```java
import java.util.Scanner;
class _14GCD {
    static int gcd(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter First Number: ");
        int num1 = scanner.nextInt();
        System.out.print("Enter Second Number: ");
        int num2 = scanner.nextInt();

        // compute GCD
        int result = gcd(num1, num2);

        System.out.println("GCD of "+num1+" and "+num2+" is: "+
           result);
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _14GCD.java
Enter First Number: 18
Enter Second Number: 12
GCD of 18 and 12 is: 6
```

Problem 15: String Palindrome

Task: Write a program to check if a given string is a palindrome.

Code:

```java
import java.util.Scanner;
class _15StringPalindrome {

    static boolean isPalindrome(String str, int left, int right)
       {
        if (left >= right) {
            return true;
        }

        if (str.charAt(left) != str.charAt(right)) {
            return false;
        }
```

```
12          return isPalindrome(str, left+1, right-1);
13      }
14
15      public static void main(String[] args) {
16          Scanner scanner = new Scanner(System.in);
17          System.out.print("Enter a String: ");
18          String str = scanner.nextLine();
19          str = str.toLowerCase();
20
21          if(isPalindrome(str, 0, str.length()-1)) {
22              System.out.println(str + " is a String Palindrome");
23          }
24          else {
25              System.out.println(str + " is not a String Palindrome
                    ");
26          }
27      }
28  }
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _15StringPalindrom.java
Enter a String: java
java is not a String Palindrome

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _15StringPalindrom.java
Enter a String: amma
amma is a String Palindrome
```

Problem 16: ASCII lower case values

Task: Write a program to print the ASCII values of all lowercase alphabets.

Code: —

```
1   class _16LowerASCII {
2       public static void main(String[] args) {
3           System.out.println("Lower Case ASCII values: ");
4           for (char ch = 'a'; ch<='z'; ch++) {
5               System.out.print(ch + " -> " + (int) ch + " ");
6           }
7       }
8   }
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _16LowerASCII.java
Lower Case ASCII values:
a -> 97 b -> 98 c -> 99 d -> 100 e -> 101 f -> 102 g -> 103 h -> 104 i -> 105
 j -> 106 k -> 107 l -> 108 m -> 109 n -> 110 o -> 111 p -> 112 q -> 113 r ->
 114 s -> 115 t -> 116 u -> 117 v -> 118 w -> 119 x -> 120 y -> 121 z -> 122
```

Problem 17: Average of List of Numbers

Task: 17. Write a program to calculate the average of a list of numbers.

Code:

```java
import java.util.Scanner;
class _17AverageCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the count of numbers: ");
        int n = scanner.nextInt();

        double sum = 0;

        for(int i=1; i<=n; i++) {
            double num = scanner.nextDouble();
            sum += num;
        }

        double avg = sum/n;

        System.out.println("Average = "+avg);
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _17AverageCalculator.java
Enter the count of numbers: 5
24
25
4
8
43
Average = 20.8
```

Problem 18: Check Leap Year

Task: Write a program to check if a given year is a leap year.

Code:

```java
import java.util.Scanner;

class _18LeapYear {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Emter a year: ");
        int year = scanner.nextInt();

        if ((year%400==0) || ((year%4==0) && (year%100!=0)))
            System.out.println(year + " is a Leap Year");
        else
```

```
            System.out.println(year + " is not a Leap Year");
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _18LeapYear.java
Emter a year: 2024
2024 is a Leap Year

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _18LeapYear.java
Emter a year: 2025
2025 is not a Leap Year
```

Problem 19: 10 Natural Numbers in reverse order

Task: Write a program to print the first 10 natural numbers in reverse order.

Code: —

```
class _19LastNatural {
    public static void main(String[] args) {
        System.out.println("First 10 Natural Numbers in reversed
            order : ");

        for(int i=10; i>=1; i--) {
            System.out.print(i + " ");
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _19LastNatural.java
First 10 Natural Numbers in reversed order :
10 9 8 7 6 5 4 3 2 1
```

Problem 20: Sum of First 50 Natural Numbers

Task: Write a program to find the sum of the first 50 natural numbers.

Code: —

```
class _20Sum50 {
    public static void main(String[] args) {
        int sum = 0;

        for (int i=1; i<=50; i++) {
            sum += i;
        }
        System.out.println("Sum of first 50 natural numbers: "+
            sum);
```

```
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\2>java _20Sum50.java
Sum of first 50 natural numbers: 1275
```

Problem 21: Factorial Numbers 1 to 10

Task: Write a program to print the factorial of numbers from 1 to 10.

Code: —

```
class _21Factorial {
    public static void main(String[] args) {
        System.out.println("Factorial of Numbers from 1 to 10: ")
           ;

        long factorial = 1;
        for (int i=1; i<=10; i++) {
            factorial *= i;
            System.out.println(i + "! = " + factorial);
            }
        }
    }
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _21Factorial.java
Factorial of Numbers from 1 to 10:
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

Problem 22: String Palindrome using loop two pointer method

Task: Write a program to check if a given string is a palindrome using a loop.

Code: —

```
import java.util.Scanner;
class _15StringPalindromeLoop {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a String: ");
```

```
        String str = scanner.nextLine();
        str = str.toLowerCase();

        int left = 0, right = str.length() - 1;
        boolean isPalindrome = true;
        // two pointer method
        while (left < right) {
            if (str.charAt(left) != str.charAt(right)){
                isPalindrome = false;
                break;
            }
            left++;
            right--;
        }

        if(isPalindrome) {
            System.out.println(str + " is a String Palindrome");
        }
        else {
            System.out.println(str + " is not a String Palindrome
               ");
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _22StringPalindromeLoop.java
Enter a String: java
java is not a String Palindrome

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _22StringPalindromeLoop.java
Enter a String: level
level is a String Palindrome
```

Problem 23: Sum of Squares of Numbers 1 to 10

Task: Write a program to calculate the sum of the squares of numbers from 1 to 10.

Code: —

```
class _23SumSquares {
    public static void main(String[] args) {

        int sum = 0;
        for (int i=1; i<=10; i++) {
            sum += (int) Math.pow(i, 2);
        }
        System.out.println("Sum of first 10 square numbers: " +
           sum);
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _23SumSquares.java
Sum of first 10 square numbers: 385
```

Problem 24: Even Numbers 1 to 100

Task: Write a program to print even numbers between 1 and 100.

Code: —

```java
class _24Even100 {
    public static void main(String[] args) {
        for (int i=2; i<=100; i++) {
            if (i%2==0)
                System.out.print(i + " ");
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _24Even100.java
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52
 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100
```

Problem 25: Sum of Odd Numbers 1 to 50

Task: Write a program to find the sum of all odd numbers between 1 and 50.

Code: —

```java
class _22SumOdd50 {
    public static void main(String[] args) {
        int sum = 0;
        for (int i=1; i<= 50; i++) {
            if (i%2!=0)
                sum += i;
        }
        System.out.print("Sum of 50 Odd Numbers: " + sum);
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _25SumOdd50.java
Sum of 50 Odd Numbers: 625
```

Problem 26: Check Perfect Number

A **perfect number** is a positive integer equal to the sum of its proper positive divisors (all divisors excluding the number itself). The first perfect number is 6, as $1+2+3=6$. Other known perfect numbers include 28, 496, and 8128.

All known perfect numbers are even, and their form is described by the *Euclid–Euler theorem*:

$$2^{p-1}(2^p-1)$$

where 2^p-1 is a *Mersenne prime*.

Task: Write a program to check if a given number is a perfect number.

Code:

```java
import java.util.Scanner;

class _26PerfectNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a Number: ");
        int num = scanner.nextInt();

        // int sum = 0;

        // // find divisors and add

        // for (int i=1; i<=num/2; i++) {
        //      if (num % i == 0)
        //              sum += i;
        // }

        // Efficient Java Implementation (O( n ))
        if (num <= 1){
            System.out.println(num + " is not a Perfect Number");
        }

        int sum = 1; // 1 is always a divisor

        // Check divisors up to sqrt(num)

        for (int i=2; i*i <= num; i++) {
            if (num % i == 0) {
                sum += i;
                if (i != num / i) {
                    sum += num / i; // add the pair divisor
                }
            }
        }

        // check perfect number
        if (num == sum && num > 0)
            System.out.println(num + " is a Perfect Number");
```

```
        else
            System.out.println(num + " is not a Perfect Number");
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _26PerfectNumber.java
Enter a Number: 6
6 is a Perfect Number

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _26PerfectNumber.java
Enter a Number: 25
25 is not a Perfect Number

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _26PerfectNumber.java
Enter a Number: 28
28 is a Perfect Number
```

Problem 27: Upper Case ASCII values

Task: Write a program to print the ASCII values of all uppercase alphabets.

Code: —

```
class _27UpperASCII {
    public static void main(String[] args) {
        for (char ch='A'; ch <= 'Z'; ch++) {
            System.out.print(ch + " -> " + (int) ch + " ");
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _27UpperASCII.java
A -> 65 B -> 66 C -> 67 D -> 68 E -> 69 F -> 70 G -> 71 H -> 72 I -> 73 J -> 74
K -> 75 L -> 76 M -> 77 N -> 78 O -> 79 P -> 80 Q -> 81 R -> 82 S -> 83 T -> 84
U -> 85 V -> 86 W -> 87 X -> 88 Y -> 89 Z -> 90
```

Problem 28: Product of the digits of a number

Task: Write a program to calculate the product of the digits of a given number.

Code: —

```
import java.util.Scanner;
class _28ProductDigit {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a Number: ");
        int num = scanner.nextInt();

        int n = num;
        int product = 1;
```

```
10
11          while (n != 0) {
12              int digit = n % 10;
13              product *= digit;
14              n /= 10;
15          }
16          System.out.println("Product of digits " + num + " is: "+
                product);
17      }
18  }
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _28ProductDigit.java
Enter a Number: 2425
Product of digits 2425 is: 80
```

Problem 29: Check Strong Number

Task: Write a program to check if a given number is a strong number.

Code: —

```
1   import java.util.Scanner;
2
3   class _29StrongNumber {
4
5       // Method to calculate factorial
6       static int factorial(int n) {
7           int fact = 1;
8           for (int i = 1; i <= n; i++) {
9               fact *= i;
10          }
11          return fact;
12      }
13
14      public static void main(String[] args) {
15          Scanner scanner = new Scanner(System.in);
16
17          System.out.print("Enter a number: ");
18          int num = scanner.nextInt();
19
20          int temp = num;
21          int sum = 0;
22
23          // Calculate sum of factorial of digits
24          while (temp != 0) {
25              int digit = temp % 10;
26              sum += factorial(digit);
27              temp /= 10;
28          }
```

```java

        // Check if strong number
        if (sum == num) {
            System.out.println(num + " is a Strong Number.");
        } else {
            System.out.println(num + " is NOT a Strong Number.");
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _29StrongNumber.java
Enter a number: 145
145 is a Strong Number.

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _29StrongNumber.java
Enter a number: 143
143 is NOT a Strong Number.
```

Problem 30: Sum of Cubes 1 to 10

Task: Write a program to calculate the sum of the cubes of numbers from 1 to 10. You can also use the formula for the sum of the first n cubes:

$$1^3 + 2^3 + \cdots + n^3 = \left(\frac{n(n+1)}{2}\right)^2$$

Code: —

```java
class _30SumCubes {
    public static void main(String[] args) {
        long total = 0;
        for (int i=1; i<=10; i++) {
            total += Math.pow(i, 3);
        }
        System.out.println("Sum of 1 to 10 cubes = " + total);
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\3>java _30SumCubes.java
Sum of 1 to 10 cubes = 3025
```

Problem 31: Sum of Prime Numbers 1 to 100

Task: Write a program to find the sum of all prime numbers between 1 and 100.

Code:

```java
class _31PrimeNumbers {

    public static void main(String[] args) {
        System.out.println("Prime numbers from 1 to 100 are:");

        int sum = 0;
        for (int i = 2; i <= 100; i++) {
            boolean isPrime = true;

            // This is an optimized approach, as a number's
               factors repeat after its square root.
            for (int j = 2; j <= Math.sqrt(i); j++) {
                // If the number is divisible by any 'j', it's
                   not prime.
                if (i % j == 0) {
                    isPrime = false;
                    break;
                }
            }
            if (isPrime) {
                System.out.print(i + " ");
                sum += i;
            }
        }
        System.out.println("Sum of 1 to 100 primes = " + sum);
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _31PrimeNumbers.ja
va
Prime numbers from 1 to 100 are:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
Sum of 1 to 100 primes = 1060
```

Problem 32: Check Pangram

Task: Write a program to check if a given string is a pangram.

Code:

```java
import java.util.Scanner;
class _32Panagram {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a String: ");
        String str = scanner.nextLine().toLowerCase();
        String alphabet = "abcdefghijklmnopqrstuvwxyz";

        int count = 0;
```

```
10
11          for (int i = 0; i<26; i++) {
12              if (str.indexOf(alphabet.charAt(i)) != -1) {
13                  count++;
14              }
15          }
16
17          if (count == 26)
18              System.out.println("Given String is a Panagram");
19          else
20              System.out.println("Not a Panagram");
21      }
22  }
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _32Panagram.java
Enter a String:
The quick brown fox jumps over the lazy dog.
Given String is a Panagram

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _32Panagram.java
Enter a String:
Hello Sony
Not a Panagram
```

Problem 33: Factorial Numbers 1 to 10

Task: Write a program to find the factorial of numbers from 1 to 10.

Code: —

```
1   class _21Factorial {
2       public static void main(String[] args) {
3           System.out.println("Factorial of Numbers from 1 to 10: ")
                ;
4
5           long factorial = 1;
6           for (int i=1; i<=10; i++) {
7               factorial *= i;
8               System.out.println(i + "! = " + factorial);
9               }
10          }
11          }
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _33Factorial.java
Factorial of Numbers from 1 to 10:
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

Problem 34: Odd Numbers 1 to 100

Task: Write a program to print odd numbers between 1 and 100.

Code:

```java
class _34OddNumbers {
    public static void main(String[] args) {
        for(int i=1; i<=100; i++) {
            if (i%2 != 0) {
                System.out.print(i + " ");
            }
        }
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _34OddNumb
ers.java
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45
 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87
 89 91 93 95 97 99
```

Problem 35: Check Perfect Square

Task: Write a program to check if a given number is a perfect square.

Code:

```java
import java.util.Scanner;
class _35PerfectSquare {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a Number: ");
        int num = scanner.nextInt();
        int i;
        // negative numbers are not perfect square
        // zero is a perfect square
        for (i=0; i*i <= num; i++) {
            if (i*i == num) {
                System.out.println(num + " is a Perfect Square");
```

```
                    break;
                }
            }
            if (i*i > num) {
                System.out.println(num + " is not a Perfect Square");
            }
        }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _35PerfectSquare.java
Enter a Number: 0
0 is a Perfect Square

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _35PerfectSquare.java
Enter a Number: 25
25 is a Perfect Square

V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _35PerfectSquare.java
Enter a Number: 24
24 is not a Perfect Square
```

Problem 36: Sum of the digits

Task: Write a program to find the sum of the digits of a given number until the sum is a single digit.

Code: —

```
import java.util.Scanner;

class _36SinglDigitRoot {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int num = sc.nextInt();

        // Reduce until single digit
        int result = num;
        while (result >= 10) {
            result = sumOfDigits(result);
        }

        System.out.println("The single digit sum is: " + result);
    }

    public static int sumOfDigits(int n) {
        int sum = 0;
        while (n > 0) {
            sum += n % 10; // add last digit
            n /= 10;       // remove last digit
        }
```

```
        return sum;
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _36SinglDigitRoot.java
Enter a number: 2425
The single digit sum is: 4
```

Problem 37: Square Pattern

Task: Square Pattern

Code: —

```
class _37SquarePattern {
    public static void main(String[] args) {
        for (int i=1; i<=5; i++) {
            System.out.println();
            for (int j=1; j<=5; j++) {
                System.out.print(" * ");
            }
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _37SquarePattern.java

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

Problem 38: Right Angled Triangle

Task: Right Angled Triangle

Code: —

```
class _38RightAngledTriangle {
    public static void main(String[] args) {
        for (int i=1; i<=5; i++) {
            System.out.println();
            for (int j=1; j<=i; j++) {
                System.out.print("* ");
            }
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _38RightAngledTriangle.java

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

Problem 39: Inverted Right Triangle

Task: Inverted Right Triangle

Code: —

```java
class _39InvertedRightAngledTriangle {
    public static void main(String[] args) {
        for (int i=5; i>=1; i--) {
            System.out.println();
            for (int j=1; j<=i; j++) {
                System.out.print("* ");
            }
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _39InvertedRightAngledTriangle.java

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

Problem 40: Pyramid

Task: Pyramid

Code: —

```java
class _40Pyramid {
    public static void main(String[] args) {

        int n = 5;
        for(int i=1; i<=n; i++) {
            for(int j=i; j<n; j++) System.out.print(" ");
            for(int j=1; j<=i; j++) System.out.print("* ");
            System.out.println();
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\4>java _40Pyramid.java
    *
   * *
  * * *
 * * * *
* * * * *
```

Problem 41: Inverted Pyramid

Task: Inverted Pyramid

Code: —

```
class _41InvertedPyramid {
    public static void main(String[] args) {

        int n = 5;
        for(int i=n; i>=1; i--) {
            for(int j=i; j<n; j++) System.out.print(" ");
            for(int j=1; j<=i; j++) System.out.print("* ");
            System.out.println();
        }

    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\5>java _41InvertedPyramid.java
* * * * *
 * * * *
  * * *
   * *
    *
```

Problem 42: Diamond

Task: Diamond

Code: —

```
class _42Diamond {
    public static void main(String[] args) {

        int n=3;
        // upper
        for(int i=1; i<=n; i++) {
            for(int j=i; j<n; j++) System.out.print(" ");
            for(int j=1; j<=i; j++) System.out.print("* ");
            System.out.println();
        }
        // lower
```

```
        for(int i=n-1; i>=1; i--) {
            for(int j=i; j<n; j++) System.out.print(" ");
            for(int j=1; j<=i; j++) System.out.print("* ");
            System.out.println();
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\5>java _42Diamond.java
  *
 * *
* * *
 * *
  *
```

Problem 43: Hallow Square

Task: Hallow Square

Code: —

```
class _43HollowSquare {
    public static void main(String[] args) {
        int n=5;
        for(int i=1; i<=n; i++) {
            for(int j=1; j<=n; j++) {
                if(i==1 || i==n || j==1 || j==n)
                    System.out.print("* ");
                else
                    System.out.print("  ");
                    }
            System.out.println();
        }
    }
}
```

Output: —

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\5>java _43HollowSquare.java
* * * * *
*       *
*       *
*       *
* * * * *
```

Problem 44: Floyd's Triangle

Task: Floyd's Triangle

Code:

```java
class _44FloydTriangle {
    public static void main(String[] args) {
        int n=5, num=1;
        for (int i=1; i<=n; i++) {
            for (int j=1; j<=i; j++) {
                System.out.print(num++ + " ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\5>java _44FloydTriangle.java
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

Problem 45: Pascal Triangle

Task: Pascal Triangle

Code:

```java
class _45PascalTriangle {
    public static void main(String[] args) {
        int n = 4;
        for (int i=0; i<n; i++) {
            for (int j=0; j<n-i; j++) System.out.print("   ");
            int num = 1;
            for (int j=0; j<=i; j++) {
                System.out.print(num + "    ");
                num = num * (i-j)/(j+1);
            }
            System.out.println();
        }
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\5>java _45PascalTriangle.java
        1
      1   1
    1   2   1
  1   3   3   1
```

Problem 46: Number Triangle

Task: Number Triangle

Code:

```java
class _46NumberTriangle {
    public static void main(String[] args) {
        for (int i=1; i<=4; i++) {
            for (int j=1; j<=i; j++) {
                System.out.print(j + " ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
V:\CDAC\3_OOP_Java\1_Assignments\4_Assignment\5>java _46NumberTriangle.java
1
1 2
1 2 3
1 2 3 4
```