Task to Do week 1

Friday – 10/11/2023

**1. Write a Java Program to find the Compound interest based on the formula below :**

**Compound Interest = P(1 + R/100)t**

Where,

P is principal amount

R is the rate and

T is the time span

Example:

Input : Principal (amount): 1200

Time: 2

Rate: 5.4

Output : Compound Interest = 1333.099243

*public class CompoundInterestCalculator {*

*public static void main(String[] args) {*

*// Principal amount*

*double p = 1200;*

*// Rate of interest*

*double r = 5.4;*

*// Time span in years*

*double t = 2;*

*// Compound interest calculation using the formula: P \* (1 + R/100)^T*

*// Math.pow is used to calculate the power (exponentiation) in the formula*

*double compoundInterest = p \* Math.pow(1 + r / 100, t);*

*// Display the calculated compound interest*

*System.out.println("Compound Interest = " + compoundInterest);*

*}*

*}*

**2. Write a Java Program to Find Factorial of a number.**

**Factorial of a non-negative integer, is multiplication of all integers smaller than or equal to n. In this article, we will learn how to write a program for the factorial of a number in Java.**

**Formulae for Factorial**

n! = n \* (n-1) \* (n-2) \* (n-3) \* ........ \* 1

Sample Output:

6! == 6\*5\*4\*3\*2\*1 = 720.

5! == 5\*4\*3\*2\*1 = 120

4! == 4\*3\*2\*1 = 24

*public class FactorialCalculator {*

*public static void main(String[] args) {*

*Scanner scanner = new Scanner(System.in);*

*System.out.print("Enter a number: ");*

*int number = scanner.nextInt();*

*long factorial = calculateFactorial(number);*

*System.out.println("Factorial of " + number + " is: " + factorial);*

*scanner.close();*

*}*

*private static long calculateFactorial(int n) {*

*if (n < 0) {*

*throw new IllegalArgumentException("Factorial is not defined for negative numbers");*

*}*

*if (n == 0 || n == 1) {*

*return 1;*

*} else {*

*return n \* calculateFactorial(n - 1);*

*}*

*}*

*}*

**3. Given two integers m and n. The goal is simply to swap their values in the memory block and writing the java code demonstrating approaches.**

**Example:**

Input : m=9, n=5

Output : m=5, n=9

Input : m=15, n=5

Output : m=5, n=15

Here 'm' and 'n' are integer value

*public class SwapDifferent {*

*public static void main(String[] args) {*

*// Declare integer variables*

*int x = 8;*

*int y = 3;*

*int z;*

*// Display values before the swap*

*System.out.println("Before swapping - X: " + x + ", Y: " + y);*

*// Swap using a temporary variable*

*z = x; // Store the value of x in z*

*x = y; // Assign the value of y to x*

*y = z; // Assign the stored value of x (in z) to y*

*// Display the swapped values*

*System.out.println("After swapping - X: " + x + ", Y: " + y);*

*}*

*}*

**4. Write a Java code to find the difference between the square of the sum and the sum of the squares of the first N natural numbers as explained below :**

The square of the sum of the first ten natural numbers is (1 + 2 + ... + 10)² = 55² = 3025.

The sum of the squares of the first ten natural numbers is 1² + 2² + ... + 10² = 385.

Hence the difference between the square of the sum of the first ten natural numbers and the sum of the squares of the first ten natural numbers is 3025 - 385 = 2640.

*public class SumSquareDifference {*

*public static void main(String[] args) {*

*int n = 10; // Change this to the desired value of N*

*// Calculate the square of the sum*

*double squareOfSum = Math.pow((n \* (n + 1) / 2.0), 2);*

*// Calculate the sum of the squares*

*double sumOfSquares = (n \* (n + 1) \* (2 \* n + 1)) / 6.0;*

*// Calculate the difference*

*double difference = squareOfSum - sumOfSquares;*

*// Display the result*

*System.out.println("The difference is: " + difference);*

*}*

*}*

**5. Write a Java code to print all the Factors of a given no.**

Example :

Number = 24

Factors : 1 , 2 , 3 , 4, 6, 8, 12, 24

*public class Factors {*

*public static void main(String[] args) {*

*Scanner scanner = new Scanner(System.in);*

*// Input the number*

*System.out.print("Enter a number: ");*

*int number = scanner.nextInt();*

*// Print the factors*

*System.out.println("Factors of " + number + " are:");*

*printFactors(number);*

*scanner.close();*

*}*

*// Function to print factors of a number*

*public static void printFactors(int number) {*

*for (int i = 1; i <= number; i++) {*

*if (number % i == 0) {*

*System.out.print(i + " ");*

*}*

*}*

*}*

*}*

**6. Write a Java code to find the LCM & GCD of 2 Numbers .**

LCM (i.e. Least Common Multiple) is the largest of the two stated numbers that can be divided by both the given numbers

Example for LCM of Two Numbers

Input: LCM( 15 and 25)

Output: 75

Input: LCM( 3 and 7 )

Output: 21

GCD (i.e. Greatest Common Divisor) or HCF (i.e. Highest Common Factor) is the largest number that can divide both the given numbers.

Example for GCD of Two Numbers:

GCD of 10 and 20 is 10

GCD of 9 and 21 is 3.

*public class LCMGCD {*

*public static void main(String[] args) {*

*Scanner scanner = new Scanner(System.in);*

*// Input two numbers*

*System.out.print("Enter the first number: ");*

*int num1 = scanner.nextInt();*

*System.out.print("Enter the second number: ");*

*int num2 = scanner.nextInt();*

*// Calculate and print LCM and GCD*

*System.out.println("LCM of " + num1 + " and " + num2 + " is: " + calculateLCM(num1, num2));*

*System.out.println("GCD of " + num1 + " and " + num2 + " is: " + calculateGCD(num1, num2));*

*scanner.close();*

*}*

*// Function to calculate LCM*

*public static int calculateLCM(int num1, int num2) {*

*return (num1 \* num2) / calculateGCD(num1, num2);*

*}*

*// Function to calculate GCD using Euclidean Algorithm*

*public static int calculateGCD(int a, int b) {*

*while (b != 0) {*

*int temp = b;*

*b = a % b;*

*a = temp;*

*}*

*return a;*

*}*

*}*