**Task to do**

**Friday – 17/11/2023**

1. Write a Java code to determine if a number is perfect, abundant, or deficient based on

Nicomachus' (60 - 120 CE) classification scheme for positive integers. The Greek mathematician

Nicomachus devised a classification scheme for positive integers, identifying each as belonging

uniquely to the categories of perfect, abundant, or deficient based on their aliquot sum. The

aliquot sum is defined as the sum of the factors of a number not including the number itself. For

example, the aliquot sum of 15 is 1 + 3 + 5 = 9.

• Perfect: aliquot sum = number

6 is a perfect number because (1 + 2 + 3) = 6

28 is a perfect number because (1 + 2 + 4 + 7 + 14) = 28

• Abundant: aliquot sum > number

12 is an abundant number because (1 + 2 + 3 + 4 + 6) = 16

24 is an abundant number because (1 + 2 + 3 + 4 + 6 + 8 + 12) = 36

• Deficient: aliquot sum < number

8 is a deficient number because (1 + 2 + 4) = 7

Prime numbers are deficient

ANSWER:

*public class NumberClassification {*

*// Method to calculate the aliquot sum of a number*

*public static int aliquotSum(int num) {*

*int sum = 1; // Start with 1 as all numbers have 1 as a factor*

*for (int i = 2; i \* i <= num; i++) {*

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

*sum += i;*

*if (i \* i != num) {*

*sum += num / i;*

*}*

*}*

*}*

*return sum;*

*}*

*// Method to classify the number*

*public static void classifyNumber(int num) {*

*int sum = aliquotSum(num);*

*if (sum == num) {*

*System.out.println(num + " is a perfect number");*

*} else if (sum > num) {*

*System.out.println(num + " is an abundant number");*

*} else {*

*System.out.println(num + " is a deficient number");*

*}*

*}*

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

*int numberToCheck = 28; // Change this number to check classification*

*classifyNumber(numberToCheck);*

*}*

*}*

1. Write a Java Code to determine if a triangle is equilateral, isosceles, or scalene.

An equilateral triangle has all three sides the same length.

An isosceles triangle has at least two sides the same length. (It is sometimes specified as

having exactly two sides the same length, but for the purposes of this exercise we'll say at

least two.)

A scalene triangle has all sides of different lengths.

ANSWER:

*public class TriangleClassification {*

*public static String classifyTriangle(int side1, int side2, int side3) {*

*if (side1 <= 0 || side2 <= 0 || side3 <= 0) {*

*return "Invalid triangle: Sides cannot be zero or negative.";*

*} else if (side1 + side2 <= side3 || side2 + side3 <= side1 || side1 + side3 <= side2) {*

*return "Invalid triangle: Sum of the lengths of any two sides must be greater than the length of the third side.";*

*} else if (side1 == side2 && side2 == side3) {*

*return "Equilateral triangle: All sides are of equal length.";*

*} else if (side1 == side2 || side2 == side3 || side1 == side3) {*

*return "Isosceles triangle: At least two sides are of equal length.";*

*} else {*

*return "Scalene triangle: All sides have different lengths.";*

*}*

*}*

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

*int side1 = 5;*

*int side2 = 5;*

*int side3 = 5;*

*String triangleType = classifyTriangle(side1, side2, side3);*

*System.out.println(triangleType);*

*}*

*}*

1. Write a Java code to check if a number is an Armstrong number.

A positive integer with digits p, q, r, s…, is known as an Armstrong number of order n if the

following condition is fulfilled.

pqrs... = pn + qn + rn + sn +

ANSWER:

*public class ArmstrongNumber {*

*public static boolean isArmstrong(int number) {*

*int originalNumber = number;*

*int numOfDigits = String.valueOf(number).length();*

*int sum = 0;*

*while (number > 0) {*

*int digit = number % 10;*

*sum += Math.pow(digit, numOfDigits);*

*number /= 10;*

*}*

*return sum == originalNumber;*

*}*

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

*int num = 153; // Change this number to check if it's an Armstrong number*

*if (isArmstrong(num)) {*

*System.out.println(num + " is an Armstrong number.");*

*} else {*

*System.out.println(num + " is not an Armstrong number.");*

*}*

*}*

*}*

1. Java Program to Print Square Star Pattern

*public class SquareStarPattern {*

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

*int rows = 5; // Change the number of rows as needed*

*// Nested loop to print the square star pattern*

*for (int i = 0; i < rows; i++) {*

*for (int j = 0; j < rows; j++) {*

*System.out.print("\* ");*

*}*

*System.out.println();*

*}*

*}*

*}*