**Design Principles Documentation**

# **A)** **Modular Design Principle**

A screen shot of a computer

Description automatically generated

This ‘printName()’ function handles the printing of names, encapsulating the specific functionality in a separate method.

A computer screen shot of code

Description automatically generated

This ‘threeNum()’ function handles input gathering for three double numbers and calls another function (‘disMaxMin()’) for further processing. Demonstrates a modular approach to handling specific tasks.

A screenshot of a computer program

Description automatically generated

This ‘disMaxMin()’ function focuses on finding the smallest and largest numbers in an array. A modular approach is evident in separating this specific functionality.

A computer screen shot of code

Description automatically generated

This ‘opt3()’ function handles input gathering for two integer numbers and calls another function (‘disRangeNum()’) for further processing. Promotes modularity in task distribution.

A computer screen shot of a program code

Description automatically generated

This ‘disRangeNum’ function displays a range of numbers and calculates the sum of odd numbers within that rage. This function encapsulates a specific task, promoting modularity.

A computer screen shot of text

Description automatically generated

This ‘numTri()’ function prompts user input for three integers and checks if they form a triangle. The function focuses on this specific tasks, adhering to modular design principles.

A computer screen shot of numbers

Description automatically generated

This ‘primeOrNot()’ function prompts user input for an integer, checks if it is prime, and displays the result. Encapsulates the primality checking functionality in a modular manner.

A computer screen shot of a number

Description automatically generated

This ‘isPrime()’ function focuses on the specific task of checking if a number is prime. Demonstrates modularity by isolating this functionality.

A computer screen shot of a program code

Description automatically generated

This ‘tenNumArray()’ function prompts user input for ten integers and calls another function (‘disHiLoAvg()’) for further processing. Promotes modularity in handling different aspects of the task.

A screenshot of a computer

Description automatically generated

This ‘disHiLoAvg()’ function displays the highest, lowest, and average values from an array of integers. Encapsulates this specific statistical calculation in a modular manner.

**B)** **Single Responsibility Principle**

A screen shot of a computer

Description automatically generated

This ‘printName()’ function prints the name of the user and tutor, focusing on a single responsibility.

A computer screen shot of a number

Description automatically generated

This ‘disMaxMin(double[] nums)’ function finds the smallest and largest numbers in an array, adhering to the single responsibility principle.

A screenshot of a computer program

Description automatically generated

This ‘disRangeNum(int start, int end)’ function displays a range of numbers and calculates the sum of odd numbers within that range. Focuses on a specific responsibility.

A computer screen shot of text

Description automatically generated

This ‘numTri()’ function prompts user input for three integers and checks if they form a triangle. Adheres to the single responsibility principle by focusing on this specific task.

A computer screen shot of numbers

Description automatically generated

This ‘primeOrNot()’ function prompts user input for an integer, checks if it is prime, and displays the result. Adheres to the single responsibility principle by concentrating on primality checking.

A computer screen shot of a number

Description automatically generated

The ‘isPrime(int num)’ function focuses on solely on the task of checking if a number is prime, following the single responsibility principle.

A computer screen shot of a computer

Description automatically generated

The ‘disHiLoAvg(int[] nums)’ function displays the highest, lowest, and average values from an array of integers. Adheres to the single responsibility principle by concentrating on statistical calculations.

**C) Other Design Principle Implemented**

## **Encapsulation:**

A computer screen with numbers and symbols

Description automatically generated

The logic on how to determine if a number is prime is encapsulated within the isPrime(int num) method.