# C# Lab 05.

**Question 03.**

using System;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter 1 for addition, 2 for subtraction, 3 for multiplication, 4 for division:");

int choice = int.Parse(Console.ReadLine());

Console.WriteLine("Enter number 1:");

double num1 = double.Parse(Console.ReadLine());

Console.WriteLine("Enter number 2:");

double num2 = double.Parse(Console.ReadLine());

double result = 0;

CalculateValues calculator = new CalculateValues();

switch (choice)

{

case 1:

result = calculator.Addition(num1, num2);

break;

case 2:

result = calculator.Subtraction(num1, num2);

break;

case 3:

result = calculator.Multiplication(num1, num2);

break;

case 4:

result = calculator.Division(num1, num2);

break;

default:

Console.WriteLine("Invalid choice");

return;

}

Console.WriteLine("Your answer is: " + result);

}

}

class CalculateValues

{

public double Addition(double num1, double num2)

{

return num1 + num2;

}

public double Subtraction(double num1, double num2)

{

return num1 - num2;

}

public double Multiplication(double num1, double num2)

{

return num1 \* num2;

}

public double Division(double num1, double num2)

{

if (num2 == 0)

{

Console.WriteLine("Cannot divide by zero");

return double.NaN;

}

return num1 / num2;

}

}

**Question 04.**

using System;

public class Helper

{

private void sayHello()

{

Console.WriteLine("Hello, World!");

}

public void CallSayHello()

{

sayHello(); // This method can access the private sayHello() method within the same class.

}

}

using System;

class Program

{

static void Main(string[] args)

{

Helper helper = new Helper();

// helper.sayHello(); // This line will cause a compilation error since sayHello() is a private method and cannot be accessed from outside the Helper class.

helper.CallSayHello(); // This line will work fine, as CallSayHello() can access the private sayHello() method within the Helper class.

}

}

* We cannot directly access the private method from outside the class. However,we can indirectly access the private method of a class by calling a public method within the same class that, in turn, calls the private method. This allows for better encapsulation and control over the internal behavior of the class.

**Question 05.**

using System;

public class ArrayHelper

{

public int[] InputValues()

{

int[] arr = new int[10];

Console.WriteLine("Enter 10 values for the array:");

for (int i = 0; i < arr.Length; i++)

{

Console.Write($"Value {i + 1}: ");

arr[i] = int.Parse(Console.ReadLine());

}

return arr;

}

public int FindMinimum(int[] arr)

{

int min = arr[0];

for (int i = 1; i < arr.Length; i++)

{

if (arr[i] < min)

min = arr[i];

}

return min;

}

public int FindMaximum(int[] arr)

{

int max = arr[0];

for (int i = 1; i < arr.Length; i++)

{

if (arr[i] > max)

max = arr[i];

}

return max;

}

public double CalculateAverage(int[] arr)

{

int sum = 0;

foreach (int num in arr)

{

sum += num;

}

return (double)sum / arr.Length;

}

public int[] ReverseArray(int[] arr)

{

int[] reversedArr = new int[arr.Length];

for (int i = 0; i < arr.Length; i++)

{

reversedArr[i] = arr[arr.Length - i - 1];

}

return reversedArr;

}

}

using System;

class Program

{

static void Main(string[] args)

{

ArrayHelper arrayHelper = new ArrayHelper();

int[] arr = arrayHelper.InputValues();

int min = arrayHelper.FindMinimum(arr);

int max = arrayHelper.FindMaximum(arr);

double average = arrayHelper.CalculateAverage(arr);

int[] reversedArr = arrayHelper.ReverseArray(arr);

Console.WriteLine("Minimum value: " + min);

Console.WriteLine("Maximum value: " + max);

Console.WriteLine("Average value: " + average);

Console.WriteLine("Array in reverse order:");

foreach (int num in reversedArr)

{

Console.Write(num + " ");

}

}

}