**Lab 05**

(3)

**Program.cs**

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Q\_3

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Select an operation: ");

Console.WriteLine("1.Addition");

Console.WriteLine("2.Subtraction");

Console.WriteLine("3.Multiplication");

Console.WriteLine("4.Division");

Console.Write("Enter your choice (1-4): ");

int choice = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter the first number: ");

double num1 = Convert.ToDouble(Console.ReadLine());

Console.Write("Enter the second number: ");

double num2 = Convert.ToDouble(Console.ReadLine());

CalculateValues calculator = new CalculateValues();

double result = 0;

switch(choice)

{

case 1:

result = calculator.Addition(num1,num2);

Console.WriteLine("The result of addition is: "+result);

break;

case 2:

result = calculator.Subtraction(num1, num2);

Console.WriteLine("The result of sutraction is: " + result);

break;

case 3:

result = calculator.Multiplication(num1, num2);

Console.WriteLine("The result of multiplication is: " + result);

break;

case 4:

result = calculator.Division(num1, num2);

Console.WriteLine("The result of division is: " + result);

break;

default:

Console.WriteLine("Invalid choice");

break;

}

Console.ReadLine();

}

}

}

**CalculateValues.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Q\_3

{

internal 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)

{

return num1 / num2;

}

}

}

(4)

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Q\_4

{

internal class Program

{

static void Main(string[] args)

{

SeparateClass sparateObj = new SeparateClass();

separateObj.SayHello(); //ERROR

/\*In this code the, 'SayHello()' method in the 'SeparateClass is declared as 'private'.A 'private' member can

\* only be accessed within the same class where it is defined.To access the 'SayHello()' method from the main

\* class,it needs to be declared as 'public'. \*/

Console.ReadLine();

}

}

}

**SeparateClass.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Q\_4

{

internal class SeparateClass

{

private void SayHello()

{

Console.WriteLine("Hello World!");

}

}

}

**(5)**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Q\_5

{

internal class Program

{

static void Main(string[] args)

{

int[] array = new int[10];

Console.WriteLine("Enter 10 numbers: ");

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

{

Console.Write("Number " + (i + 1) + ": ");

array[i] = Convert.ToInt32(Console.ReadLine());

}

ArrayOperations arrayOps = new ArrayOperations();

arrayOps.PerformArrayOperations(array);

Console.ReadLine();

}

}

}

**ArrayOperations.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Q\_5

{

internal class ArrayOperations

{

public void PerformArrayOperations(int[] array)

{

int minValue = FindMinimumValue(array);

int maxValue = FindMaximumValue(array);

double averageValue = FindAverageValue(array);

int[] reverseArray = ReverseArray(array);

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

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

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

Console.WriteLine("Reversed array: ");

foreach(int num in reverseArray)

{

Console.Write(num + " ");

}

}

private int FindMinimumValue(int[] array)

{

int minValue = array[0];

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

{

if (array[i] < minValue)

{

minValue = array[i];

}

}

return minValue;

}

private int FindMaximumValue(int[] array)

{

int maxValue = array[0];

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

{

if (array[i] > maxValue)

{

maxValue = array[i];

}

}

return maxValue;

}

private double FindAverageValue(int[] array)

{

int sum = 0;

foreach(int num in array)

{

sum += num;

}

double average = (double)sum / array.Length;

return average;

}

private int[] ReverseArray(int[] array)

{

int[] reverseArray = new int[array.Length];

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

{

reverseArray[i] = array[array.Length - 1 - i];

}

return reverseArray;

}

}

}