Question 03

Enter 1 for Addition

Enter 2 for Subtraction

Enter 3 for Multiplication

Enter 4 for Division

Enter Your Choice: 3

Enter Number 1: 25

Enter Number 2: 2

Your Answer is: 50

figure 01

Create the above mentioned console application and display it to the user. If user need to do an Addition user need to insert 1 as the choice. For subtraction it should be 2 etc. Your program should contain a separate class call “CalculateValues” and inside the class you should add four methods which perform four arithmetic operations. All the methods should take two parameters which are user inserted numbers. And at the end of the method return the answer out of the method. In main class if user want to do an addition call only the addition method in separate class. If user want to do a subtraction call only the subtraction method in separate class etc. And display the final answer as shown in the figure 01.

using System;

namespace ConsoleApp28

{

internal class CalculateValues

{

static void Main(string[] args)

{

Console.WriteLine("Enter 1 for Addition");

Console.WriteLine("Enter 2 for Subtraction");

Console.WriteLine("Enter 3 for Multiplication");

Console.WriteLine("Enter 4 for Division");

Console.WriteLine();

Console.Write("Enter Your Choice: ");

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

Console.Write("Enter Number 1: ");

int number1 = int.Parse(Console.ReadLine());

Console.Write("Enter Number 2: ");

int number2 = int.Parse(Console.ReadLine());

int result = 0;

switch (choice)

{

case 1:

result = CalculateValues.Addition(number1, number2);

break;

case 2:

result = CalculateValues.Subtraction(number1, number2);

break;

case 3:

result = CalculateValues.Multiplication(number1, number2);

break;

case 4:

double divisionResult = CalculateValues.Division(number1, number2);

Console.WriteLine("Your Answer is: " + divisionResult);

return;

default:

Console.WriteLine("Invalid choice!");

return;

}

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

Console.ReadLine();

}

public static int Addition(int num1, int num2)

{

return num1 + num2;

}

public static int Subtraction(int num1, int num2)

{

return num1 - num2;

}

public static int Multiplication(int num1, int num2)

{

return num1 \* num2;

}

public static double Division(double num1, double num2)

{

if (num2 == 0)

{

Console.WriteLine("Error: Division by zero is not allowed.");

return 0;

}

return num1 / num2;

}

}

}

Question 04

Add a separate class file to Console application program and create a method call private void sayHello(). Inside the method display hello world. In main class create object and try to access the sayHello() method by using the class object.

Can you access the method? Explain why?

**MyClass.cs:**

public class MyClass

{

private void sayHello()

{

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

}

}

**Program.cs:**

class Program

{

static void Main(string[] args)

{

MyClass myObject = new MyClass();

myObject.sayHello();

}

}

When you try to access the **‘sayHello()’** method using the class object **‘myObject‘** in the **‘Main’** method, you will encounter an error stating that **'MyClass.sayHello()'** **is inaccessible due to its protection level**.

The reason for this error is that the **‘sayHello()**’ method is declared as private. Private methods can only be accessed within the same class and are not visible outside of that class. Since you're trying to access the **‘sayHello()’** method from the **‘Main’** method in a different class, you cannot access it due to the private access modifier.

If you want to access the **‘sayHello()’** method from the **‘Main’** method, you need to change the access modifier of the **‘sayHello()’** method to either **‘public’** or **‘internal’**. Here's an example using the **‘public’** access modifier:

**MyClass.cs:**

public class MyClass

{

public void sayHello()

{

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

}

}

**Program.cs:**

class Program

{

static void Main(string[] args)

{

MyClass myObject = new MyClass();

myObject.sayHello();

}

}

Now, you can access the **‘sayHello()’** method from the **‘Main’** method because it's declared as **‘public’**.

Question 05

Declare a Single dimensional array with 10 elements. Input the values to the array and find the

followings,

• Minimum value.

• Maximum value.

• Average value.

• Reverse order of values.

Hint – use a method which in separate class. And call the method from main the method.

using System;

namespace ConsoleApp29

{

internal class ArrayOperations

{

static void Main(string[] args)

{

int[] array = new int[10];

Console.WriteLine("Enter 10 integer values:");

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

{

Console.Write("Enter value for element " + i + ": ");

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

}

ArrayOperations.PerformArrayOperations(array);

Console.ReadLine();

}

public static void PerformArrayOperations(int[] arr)

{

int minValue = arr[0];

int maxValue = arr[0];

int sum = 0;

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

{

if (arr[i] < minValue)

minValue = arr[i];

if (arr[i] > maxValue)

maxValue = arr[i];

sum += arr[i];

}

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

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

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

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

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

for (int i = arr.Length - 1; i >= 0; i--)

{

Console.Write(arr[i] + " ");

}

Console.WriteLine();

}

}

}