C# .Net Programming Assignment 6

- Create separate visual Studio project for each problem statement separately.
- For Business logic write separate class.
- Use Object Oriented concepts while writing the program.

1. There is one abstract class named as Marvellous which contains two abstract and three concrete methods which are used to check whether the number is strong or Armstrong or not.

Design one another class named as Numbers which inherits Marvellous abstract class and provide definitions of all abstract methods.

Note :

Armstrong number :

An Armstrong number is a number which is equal to the sum of digits raise to the power total number of digits in the number.

For example, 371 is an Armstrong number since 3**3 + 7**3 + 1**3 = 371.

Strong number:

Strong numbers are the numbers whose sum of factorial of digits is equal to the original number.

For example, 145 -> 1! + 4! + 5! = 145

```
abstract class Marvellous
{
	public int no;

	public Marvellous(int value)
	{
		no = value;
	}

	public int Power(int x, int y)
	{
		// Write logic for x^y
	}

	public int SumFactor(int no)
	{
		// Return addition of factors of given number
	}

	public int CountDigit(int no)
	{
		// Logic to return number of digits from input number
```

```
        }

        // Abstract method declarations

        public abstract boolean CheckStrong();
        public abstract boolean CheckArmstrong();
}

class Numbers : Marvellous
{
        public Number(int value ) : _________
        {
        }

        // Override method CheckStrong
        public override boolean CheckStrong()
        {
                // Logic to check whether number is strong or not
                // Use concrete methods above abstract class
        }

        // Override method CheckArmstrong
        public override boolean CheckArmstrong()
        {
                // Logic to check whether number is Armstrong or not
                // Use concrete methods above abstract class
        }
}

class Program
{
   static void Main(string[] args)
   {
           // Create object of Numbers class and call CheckStrong and CheckArmstrong
methods
   }
}
```

2. There is one abstract class named as MarvellousRange which contains two characteristics as start and end which are consider as range.
That class contains abstract methods as

SumRange() : **Returns summation of all elements from given range**
DisplayEven() : **Display all even numbers from given range**

DisplayOdd() : **Display all even numbers from given range**
DisplayPrime() : **Display all even numbers from given range**
DisplayPerfect() : **Display all even numbers from given range**

Design one another class named as MyRange which inherits MarvellousRange abstract class and provide definitions of all abstract methods.

```
abstract class MarvellousRange
{
	public int iStart;
	public int iEnd;

	public MarvellousRange(int value1, int value2)
	{
		iStart = value1;
		iEnd = value2;
	}

	// Abstract method declarations

	public abstract int SumRange();
	public abstract void DisplayEven();
	public abstract void DisplayOdd();
	public abstract void DisplayPrime();
	public abstract void DisplayPerfect();
}

class MyRange : MarvellousRange
{
	public Number(int value1, int value2 ) : ________
	{
	}

	public abstract int SumRange()
	{
		// Logic
	}

	public abstract void DisplayEven()
	{
		// Logic
	}
```

```
        public abstract void DisplayOdd()
        {
                // Logic
        }

        public abstract void DisplayPrime()
        {
                // Logic
        }

        public abstract void DisplayPerfect()
        {
                // Logic
        }
}

class Program
{
   static void Main(string[] args)
   {
                // Create object of MyRange class and call all methods
   }
}
```