C# applications for Method overloading

Polymorphism means Single name multiple behaviour.

Thera are two types of polymorphisms as

1. **Compile time polymorphism**
2. **Run time polymorphism**

We can achieve compile time polymorphism using function overloading.

In function overloading we define multiple functions having same name but different types of arguments or different number of arguments or different sequence of arguments.

Application 1 :

```
using System;

    class Marvellous
    {
        // By changing number of parameters
        public int Add(int no1, int no2)
        {
            return (no1 + no2);
        }

        public int Add(int no1, int no2, int no3)
        {
            return (no1 + no2 + no3);
        }

        // By changing type of parameters
        public int Sub(int no1, int no2)
        {
            return (no1 - no2);
        }

        public float Sub(float no1, float no2)
        {
            return (no1 - no2);
        }
        // By changing sequance of parameters
        public float AddX(int no1, float no2)
        {
            return (no1 + no2);
        }

        public float AddX(float no1, int no2)
        {
            return (no1 + no2);
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Marvellous dobj = new Marvellous();

                // We can call different overloaded methods as

            Console.WriteLine("First call {0}\n", dobj.Add(10, 1));
            Console.WriteLine("Second call {0}\n", dobj.Add(10, 1, 10));
            Console.WriteLine("Third call {0}\n", dobj.Sub(15, 4));
            Console.WriteLine("Fourth call {0}\n", dobj.Sub(10.0f,3.2f));
            Console.WriteLine("Fifth call {0}\n", dobj.AddX(15, 4.8f));
            Console.WriteLine("Sixth call {0}\n", dobj.AddX(10.2f, 3));
        }
    }
```