1. Explain Constructor in C#

Solution –

A constructor is a special method of the class which gets automatically invoked whenever an instance of the class is created. Like methods, a constructor also contains the collection of instructions that are executed at the time of Object creation. It is used to assign initial values to the data members of the same class.

Types of Constructors: -
1. Default Constructor
2. Parameterized Constructor
3. Copy Constructor
4. Private Constructor
5. Static Constructor

Default Constructor

A constructor with no parameters is called a default constructor. A default constructor has every instance of the class to be initialized to the same values. The default constructor initializes all numeric fields to zero and all string and object fields to null inside a class.

```
using System;
namespace Abhishek
{
    class DefaultConstructor
    {
        DefaultConstructor()
        {
            Console.WriteLine("Abhishek Majumdar");
        }
        public static void Main(string [] abhi)
        {
            DefaultConstructor obj=new DefaultConstructor();
            Console.ReadKey();
        }
    }
}
```


Abhishek Majumdar
Contact – 7001916946 | Email – webdevabhi4@gmail.com

Parameterized Constructor

A constructor have at least one parameter is called a parameterized constructor. It can initialize each instance of the class to different values.

```
using System;
namespace Abhishek
{
    class ParameterizedConstructor
    {
        int x,y;
        ParameterizedConstructor(int a,int b)
        {
            this.x=a;
            this.y=b;
        }
        public static void Main(string [] abhi)
        {
            ParameterizedConstructor obj=new ParameterizedConstructor(5,10);
            int res=obj.x+obj.y;
            Console.WriteLine("Res - "+res);
            Console.ReadKey();
        }
    }
}
```

Copy Constructor

This constructor will creates an object by copying variables from another object. Its main use is to initialize a new instance to the values of an existing instance.

```
using System;
namespace Abhishek
{
    class CopyConstructor
    {
        int x,y;
        CopyConstructor(CopyConstructor p)
        {
            x=p.x;
            y=p.y;
        }
        CopyConstructor(int a,int b)
        {
```

```
            this.x=a;
            this.y=b;
        }
        public static void Main(string [] abhi)
        {
            CopyConstructor obj1=new CopyConstructor(5,10);
            CopyConstructor obj2=new CopyConstructor(obj1);
            int res=obj2.x+obj2.y;
            Console.WriteLine("Res - "+res);
            Console.ReadKey();
        }
    }
}
```

Private Constructor

If a constructor is created with private specifier is known as Private Constructor. It is not possible for other classes to derive from this class and also it's not possible to create an instance of this class.

```
using System;
namespace Abhishek
{
    class PrivateConstructor
    {
        int x,y;
        private PrivateConstructor()
        {
            Console.WriteLine("Hey... I am a Private Constructor... ");
        }
        PrivateConstructor(int a,int b)
        {
            this.x=a;
            this.y=b;
        }
        public static void Main(string [] abhi)
        {
            PrivateConstructor obj1=new PrivateConstructor(5,10);
            int res=obj1.x+obj1.y;
            Console.WriteLine("Res - "+res);
            Console.ReadKey();
        }
    }
}
```


Abhishek Majumdar
Contact – 7001916946 | Email – webdevabhi4@gmail.com

Static Constructor

Static Constructor has to be invoked only once in the class and it has been invoked during the creation of the first reference to a static member in the class. A static constructor is initialized static fields or data of the class and to be executed only once.

```
using System;
namespace Abhishek
{
    class StaticConstructor
    {
        int x,y;
        static StaticConstructor()
        {
            Console.WriteLine("Hey... I am a Copy Constructor... ");
        }
        StaticConstructor()
        {
            Console.WriteLine("Hey... I am a Default Constructor... ");
        }
        public static void Main(string [] abhi)
        {
            StaticConstructor obj1=new StaticConstructor();
            StaticConstructor obj2=new StaticConstructor();
            Console.ReadKey();
        }
    }
}
```


Abhishek Majumdar
Contact – 7001916946 | Email – webdevabhi4@gmail.com