What is constructor?

A constructor is a special method that is used to initialize an object. Constructor is invoked at the time of object creation. Constructor name must be same as its class name. Constructor must have no explicit return type.

What are the types of constructors in c#?

There are five types of constructors

1. Default constructor
2. Parameterized constructor
3. Copy constructor
4. Static constructor
5. Private constructor

Different between Constructors and Method

Constructor	Method
Constructor is used to initialize an object	Method is used to expose behavior of an object
Constructor must not have return type.	Method have or not have return type.
Constructor must be same as the class name	Method name may or may not be same as the class name
Constructor is invoke implicitly.	Method is invoke explicitly.

What is default constructor in c#?

Default Constructor: A constructor without any parameters is called a default constructor. If we do not create constructor the class will automatically call default constructor when object is created.

```
using System;
namespace DefaultConstructor_Demo
{
    public class Customer
    {
        public string firstName;
        public string lastName;
        public Customer()
        {

        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Customer custormer = new Customer();
            custormer.firstName = "Farhan";
            custormer.lastName = "Ahmed";
```

```
            Console.WriteLine("Full Name:"+custormer.firstName+ " "+ custormer.lastName);
            Console.ReadLine();
        }
    }
}
```

What is the parameterized constructor in c#?

Parameter Constructor: A constructor with at least one parameter is called a parametrized constructor.

```
using System;
namespace ParameterConstructor_Demo
{
    class ParameterConstructor
    {
        public int FirstNumber;
        public int SecondNumber;

        public ParameterConstructor(int firstNumber, int secondNumber)
        {
            FirstNumber = firstNumber;
            SecondNumber = secondNumber;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            ParameterConstructor p = new ParameterConstructor(10, 20);
            int Result = p.FirstNumber + p.SecondNumber;

            Console.WriteLine("Total:" + Result);
            Console.ReadLine();
        }
    }
}
```

What is copy constructor in c#?

Copy Constructor: The constructor which creates an object by copying variables from another object is called a copy constructor.

```
using System;
namespace CopyConstructor_Demo
{
    public class Employee
    {
       public string firstName;
       public string lastName;
       public string position;
       public int salary;
        public Employee()
        {

        }
        // Copy constructor.
        public Employee(Employee employee)
        {
```

```
            firstName = employee.firstName;
            lastName  = employee.lastName;
            position  = employee.position;
            salary    = employee.salary;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Employee emp = new Employee();
            Employee emp1 = new Employee(emp);

            Console.WriteLine("Enter your first name:");
            emp1.firstName = Convert.ToString(Console.ReadLine());
            Console.WriteLine("Enter your last name:");
            emp1.lastName = Convert.ToString(Console.ReadLine());
            Console.WriteLine("Enter your position:");
            emp1.position = Convert.ToString(Console.ReadLine());
            Console.WriteLine("Enter your salary:");
            emp1.salary = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("First Name:" + emp1.firstName);
            Console.WriteLine("Last Name:" + emp1.lastName);
            Console.WriteLine("Position:" + emp1.position);
            Console.WriteLine("Salary:" + emp1.salary);
        }
    }
}
```

What is static constructor in c#?

Static Constructor: A static constructor is used to initialize any static data, or to perform a particular action that needs to be performed once only. It is called automatically before the first instance is created or any static members are referenced.

Characteristic of static constructor

1. A static constructor does not take any access modifiers.
2. A static constructor does not have parameter.
3. A static constructor is called automatically to initialize the class before the first instance is created or any static members are referenced.
4. A static constructor cannot be called directly.
5. The user has no control on when the static constructor is executed in the program.
6. A typical use of static constructors is when the class is using a log file and the constructor is used to write entries to this file.
7. A class can have only one static constructor.
8. It can access only static members of class.

```
using System;
namespace StaticConstructor_Demo
{
    public class Customer
    {
        public static string firstName;
        public static string lastName;
        private Customer()
```

```
        {
            Console.WriteLine("I am Platinum Member");
        }
        public Customer(string FirstName,string LastName)
        {
            firstName = FirstName;
            lastName = LastName;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            // The following comment line will throw an error because constructor is
inaccessible
            //Customer customer = new Customer();
            Customer customer1 = new Customer("Farhan","Ahmed");
            Console.WriteLine("Full Name:"+Customer.firstName +" "+Customer.lastName);
            Console.ReadLine();
        }
    }
}
```

What is private constructor in c#?

Private Constructor: A private constructor is a special instance constructor. It is generally used in classes that contain static members only. If a class has one or more private constructors and no public constructors, other classes (except nested classes) cannot create instances of this class. The use of private constructor is to serve singleton classes. A singleton class is one which limits the number of objects creation to one. Using private constructor we can ensure that no more than one object can be created at a time

1. One use of private constructor is when we have only static member.
2. It provide implementation of singleton class pattern
3. Once we provide constructor (private/public/any) the compiler will not add the no parameter public constructor to any class.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PrivateConstructor_Demo
{
    public class Candidate
    {
        private Candidate()
        {
        }
        public static int CandidateVisitedForInterview;
        public static int CountCandidate()
        {
            return ++CandidateVisitedForInterview;
        }
    }
    class Program
    {
```

```
        static void Main(string[] args)
        {
            // The following comment line will throw an error because constructor is
inaccessible
            //Candidate candidate = new Candidate();
            Candidate.CandidateVisitedForInterview = 20;
            Candidate.CountCandidate();
            Console.WriteLine("Interviewed candidates: {0}",
Candidate.CandidateVisitedForInterview);
            Console.ReadLine();
        }
    }
}
```