C# Object and Class Example

```
using System;
  public class Student
   {
      int id;//data member (also instance variable)
      String name;//data member(also instance variable)

    public static void Main(string[] args)
      {
         Student s1 = new Student();//creating an object of Student
         s1.id = 101;
         s1.name = "Sonoo Jaiswal";
         Console.WriteLine(s1.id);
         Console.WriteLine(s1.name);

      }
   }
```

C# Class Example 2: Having Main() in another class

```
using System;
  public class Student
   {
      public int id;
      public String name;
  }
  class TestStudent{
     public static void Main(string[] args)
      {
         Student s1 = new Student();
         s1.id = 101;
         s1.name = "Sonoo Jaiswal";
         Console.WriteLine(s1.id);
         Console.WriteLine(s1.name);

      }
   }
```

C# Class Example 3: Initialize and Display data through method

```
using System;
  public class Student
   {
      public int id;
      public String name;
      public void insert(int i, String n)
      {
         id = i;
         name = n;
      }
      public void display()
      {
         Console.WriteLine(id + " " + name);
      }
  }
  class TestStudent{
     public static void Main(string[] args)
      {
         Student s1 = new Student();
         Student s2 = new Student();
         s1.insert(101, "Ajeet");
         s2.insert(102, "Tom");
         s1.display();
         s2.display();

      }
   }
```

C# Default Constructor

```
using System;
  public class Employee
   {
      public Employee()
      {
         Console.WriteLine("Default Constructor Invoked");
      }
   }
  class TestEmployee{
     public static void Main(string[] args)
      {
         Employee e1 = new Employee();
         Employee e2 = new Employee();
      }
   }
```

C# Parameterized Constructor

```
using System;
  public class Employee
   {
      public int id;
      public String name;
      public float salary;
      public Employee(int i, String n,float s)
      {
         id = i;
         name = n;
         salary = s;
      }
      public void display()
      {
         Console.WriteLine(id + " " + name+" "+salary);
      }
```

```
}
class TestEmployee{
    public static void Main(string[] args)
    {
        Employee e1 = new Employee(101, "Sonoo", 890000f);
        Employee e2 = new Employee(102, "Mahesh", 490000f);
        e1.display();
        e2.display();

    }
}
```

C# Destructor

```
using System;
public class Employee
{
    public Employee()
    {
        Console.WriteLine("Constructor Invoked");
    }
    ~Employee()
    {
        Console.WriteLine("Destructor Invoked");
    }
}
class TestEmployee{
    public static void Main(string[] args)
    {
        Employee e1 = new Employee();
        Employee e2 = new Employee();
    }
}
```

C# Properties

```
using System;
  public class Employee
   {
      private string name;

      public string Name
      {
         get
         {
            return name;
         }
         set
         {
            name = value;
         }
      }
  }
  class TestEmployee{
     public static void Main(string[] args)
      {
         Employee e1 = new Employee();
         e1.Name = "Sonoo Jaiswal";
         Console.WriteLine("Employee Name: " + e1.Name);

      }
   }
```

C# Auto-Implemented Properties

```
using System;
using System.Collections.Generic;
namespace CSharpFeatures
{
    class Student
    {
        // Auto-implimented Properties
        public int ID { get; set; }
        public string Name { get; set; }
        public string Email { get; set; }
    }
    class AutoImplementedProperty
    {
        public static void Main(string[] args)
        {
            Student student = new Student();
            // Setting properties
            student.ID    = 101;
            student.Name  = "Rahul Kumar";
            student.Email = "rahul@example.com";
            // Getting properties
            Console.WriteLine(student.ID);
            Console.WriteLine(student.Name);
            Console.WriteLine(student.Email);
        }
    }
}
```

C# - Object Initializer

```
public class Student
{
    public int StudentID { get; set; }
    public string StudentName { get; set; }
    public int Age { get; set; }
    public string Address { get; set; }
}

class Program
{
    static void Main(string[] args)
    {
        Student std = new Student() { StudentID = 1,
                                      StudentName = "Bill",
                                      Age = 20,
                                      Address = "New York"
                                    };
    }
}
```

C# Static and Instance methods

```
class Car
{
  //Static method or class method
  public static void run()
  {
    Console.WriteLine("I'm Static method");
  }

  //non static method or instanse method
  public void engine()
  {
    Console.WriteLine("I'm non-static/instance method!!!");
  }

}
```

```
class Program
{
    static void Main(string[] args)
    {
        //Call static method by class name
        Car.run();
        //Call non-static method. call method by object
        Car c = new Car();
        c.engine();
    }
}
```

Method Overloading in C#

```
namespace PolymorphismDemo
{
 class Program
 {
  public void add(int a, int b)
  {
    Console.WriteLine(a + b);
  }
  public void add(float x, float y)
  {
   Console.WriteLine(x + y);
  }
  public void add(string s1, string s2)
  {
   Console.WriteLine(s1 + s2);
  }
  static void Main(string[] args)
  {
    Program obj = new Program();
    obj.add(10, 20);
    obj.add(10.5f, 20.5f);
    obj.add("pranaya", "kumar");
```

```
  Console.WriteLine("Press any key to exist.");
  Console.ReadKey();
  }
 }
}
```

Partial Classes

```
namespace PartialDemo
{
public class Employee
{
private string _firstName;
private string _lastName;
private double _salary;
private string _gender;
public string FirstName
{
get { return _firstName; }
set { _firstName = value; }
}
public string LastName
{
get { return _lastName; }
set { _lastName = value; }
}
public double Salary
{
get { return _salary; }
set { _salary = value; }
}
public string Gender
{
get { return _gender; }
set { _gender = value; }
}
```

```
public void DisplayFullName()
{
Console.WriteLine(@"Full Name is : {0} {1}", _firstName, _lastName);
}
public void DisplayEmployeeDetails()
{
Console.WriteLine("Employee Details : ");
Console.WriteLine(@"First Name : {0}", _firstName);
Console.WriteLine(@"Last Name : {0}", _lastName);
Console.WriteLine(@"Gender : {0}", _gender);
Console.WriteLine(@"Salary : {0}", _salary);
}
}
}
namespace PartialDemo
{
class Program
{
static void Main(string[] args)
{
Employee emp = new Employee();
emp.FirstName = "Pranaya";
emp.LastName = "Rout";
emp.Salary = 100000;
emp.Gender = "Male";
emp.DisplayFullName();
emp.DisplayEmployeeDetails();
Console.WriteLine("Press any key to exist.");
Console.ReadKey();
}
}
}
```

C# Command Line Arguments

```
using System;
namespace CSharpProgram
{
   class Program
   {
      // Main function, execution entry point of the program
      static void Main(string[] args) // string type parameters
      {
         // Command line arguments
         Console.WriteLine("Argument length: "+args.Length);
         Console.WriteLine("Supplied Arguments are:");
         foreach (Object obj in args)
         {
            Console.WriteLine(obj);
         }
      }
   }
}
```