Q.1 Discuss about Class and Object In C# Pragramming.

Ans :

Class:

In C#, class is a group of similar objects. It is a template from which objects are created. It can have fields, methods, constructors etc. The Class keyword is used to define a class in C# Programming.

Let's see an example of C# class that has two fields only.

```
public class Student
 {
    int id;//field or data member
    String name;//field or data member
 }
```

Object :

In C#, Object is a real world entity, for example, chair, car, pen, mobile, laptop etc.
In other words, object is an entity that has state and behavior. Here, state means data and behavior means functionality. Object is a runtime entity, it is created at runtime. Object is an instance of a class. All the members of the class can be accessed through object.
Let's see an example to create object using new keyword.

```
Student s1 = new Student();//creating an object of Student
```

In this example, Student is the type and s1 is the reference variable that refers to the instance of Student class. The new keyword allocates memory at runtime.

.....

Example1:

Let's see an example of class that has two fields: id and name. It creates instance of the class, initializes the object and prints the object value.

```
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);
```

```
    }
}
```

Example 2:

Let's see another example of class where we are having Main() method in another class. In such case, class must be public.

```
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);

     }
 }
```

Example 3:

Let's see another example of C# class where we are initializing and displaying object 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();

        }
    }
```

Example 4:

C# Class Example 4: Store and Display Employee Information

```
    using System;
      public class Employee
       {
          public int id;
          public String name;
          public float salary;
          public void insert(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();
          Employee e2 = new Employee();
          e1.insert(101, "Sonoo",890000f);
             e2.insert(102, "Mahesh", 490000f);
             e1.display();
             e2.display();

          }
       }
```

Q.3 Define Methods in C # Programming.

Ans:

Method:

A **method** is a block of code which only runs when it is called. We can pass data, known as parameters, into a method. Methods are used to perform certain actions, and they are also known as **functions**

A method is defined with the name of the method, followed by parentheses **()**. C# provides some pre-defined methods, which we already are familiar with, such as `Main()`, but we can also create our own methods to perform certain actions.

The syntax for defining a method in C# is as follows –

```
<Access Specifier> <Return Type> <Method Name>(Parameter List)
 {
   Method Body
}
```

Following are the various elements of a method −

- **Access Specifier** − This determines the visibility of a variable or a method from another class.
- **Return type** − A method may return a value. The return type is the data type of the value the method returns. If the method is not returning any values, then the return type is **void**.
- **Method name** − Method name is a unique identifier and it is case sensitive. It cannot be same as any other identifier declared in the class.
- **Parameter list** − Enclosed between parentheses, the parameters are used to pass and receive data from a method. The parameter list refers to the type, order, and number of the parameters of a method. Parameters are optional; that is, a method may contain no parameters.
- **Method body** − This contains the set of instructions needed to complete the required activity.

Call a Method:

To call (execute) a method, write the method's name followed by two parentheses **()** and a semicolon**;**

Example:

```
using System;

class Methodexp

{

static void MyMethod()

{

  Console.WriteLine("I just got executed!");
```

```
}
static void Main(string[] args)
{
  MyMethod();
}
}
```

……..

Q4. Define Constructor with example in C# language.

Ans:

In C#, constructor is a special method which is invoked automatically at the time of object creation. It is used to initialize the data members of new object generally. The constructor in C# has the same name as class or struct.

There can be two types of constructors in C#.

- Default constructor
- Parameterized constructor

Default constructor:

A constructor which has no argument is known as default constructor. It is invoked at the time of creating object.

Example:

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

Parameterized constructor:

A constructor which has parameters is called parameterized constructor. It is used to provide different values to distinct objects.

Example:

```
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();

       }
   }
```

……

Q. 5. Define the used of New operator and this keyword.

Ans :

this keyword:

In c# programming, this is a keyword that refers to the current instance of the class. There can be 3 main usage of this keyword in C#.

- It can be used **to refer current class instance variable**. It is used if field names (instance variables) and parameter names are same, that is why both can be distinguish easily.
- It can be used **to pass current object as a parameter to another method**.
- It can be used **to declare indexers**.

Example:

```
using System;
  public class Employee
   {
     public int id;
      public String name;
      public float salary;
      public Employee(int id, String name,float salary)
      {
         this.id = id;
         this.name = name;
         this.salary = salary;
      }
     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();

      }
   }
```

new Operator:

Use the new keyword to create an instance of the array. The new operator is used to create an object or instantiate an object. Here in the example an object is created for the class using the new.

The following is an example.

```
Calculate c = new Calculate();
```

We can also use the new keyword to create an instance of the array.

```
double[] points = new double[10];
```

The new keyword is also used to create object of a collection.

```
SortedList sl = new SortedList(); // SortedList

List<string> myList = new List<string>()  // List
```

.....