

C# Object and Class

Since C# is an object-oriented language, program is designed using objects and classes in C#.

C# 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.

C# 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.

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

1. **public class** Student
2. {
3. **int** id;//field or data member
4. String name;//field or data member
5. }

C# Object and Class Example

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.

1. **using** System;
2. **public class** Student
3. {
4. **int** id;//data member (also instance variable)
5. String name;//data member(also instance variable)
6. }
7. **public static void** Main(**string**[] args)
8. {

```
9.         Student s1 = new Student();//creating an object of Student
10.         s1.id = 101;
11.         s1.name = "Zain";
12.         Console.WriteLine(s1.id);
13.         Console.WriteLine(s1.name);
14.
15.     }
16. }
```

Output:

```
101
Zain
```

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

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

```
1. using System;
2. public class Student
3. {
4.     public int id;
5.     public String name;
6. }
7. class TestStudent{
8.     public static void Main(string[] args)
9.     {
10.         Student s1 = new Student();
11.         s1.id = 101;
12.         s1.name = "Zain";
13.         Console.WriteLine(s1.id);
14.         Console.WriteLine(s1.name);
15.
16.     }
17. }
```

Output:

```
101
Zain
```

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

Let's see another example of C# class where we are initializing and displaying object through method.

```
1. using System;
2. public class Student
3. {
4.     public int id;
5.     public String name;
6.     public void insert(int i, String n)
7.     {
8.         id = i;
9.         name = n;
10.    }
11.    public void display()
12.    {
13.        Console.WriteLine(id + " " + name);
14.    }
15. }
16. class TestStudent{
17.     public static void Main(string[] args)
18.     {
19.         Student s1 = new Student();
20.         Student s2 = new Student();
21.         s1.insert(101, "Ali");
22.         s2.insert(102, "Ahmed");
23.         s1.display();
24.         s2.display();
25.
26.     }
27. }
```

Output:

```
101 Ali
102 Ahmed
```

C# Class Example 4: Store and Display Employee Information

```
1. using System;
2. public class Employee
3. {
4.     public int id;
5.     public String name;
6.     public float salary;
7.     public void insert(int i, String n,float s)
8.     {
9.         id = i;
10.        name = n;
11.        salary = s;
12.    }
13.    public void display()
14.    {
15.        Console.WriteLine(id + " " + name+" "+salary);
16.    }
17. }
18. class TestEmployee{
19.     public static void Main(string[] args)
20.     {
21.         Employee e1 = new Employee();
22.         Employee e2 = new Employee();
23.         e1.insert(101, "Zain Khan",890000f);
24.         e2.insert(102, "Ali Ahmed", 490000f);
25.         e1.display();
26.         e2.display();
27.
28.     }
29. }
```

Output:

```
101 Zain Khan 890000
102 Ali Ahmed 490000
```