# Object Class

This is the ultimate base class of all classes in the .NET Framework; it is the root of the type hierarchy.

The Object class has five methods:

- GetType
- Equals
- ReferenceEquals
- ToString
- GetHashCode

## GetType

The Type class can be used to get type information from the object. The GetType method of an object returns a type object, which you can use to get information on an object such as its name, namespace, base type and so on.

## Equal

The Equal method of the Object class can compare two objects.

## ReferenceEqual

The Reference method can compare two objects instances.

## ToString

The ToString method of the Object class converts a type to a string type.

## GetHashCode

The GetHashCode method returns the hashcode of an object. To return a hashcode for a type, you must override the GetHashCode method. An integer value is returned which represents whether and object is available in a hashtable.

**Application :**

**Application which demonstrates use of all methods from Object class.**

```
using System;

    class Base
    {
        public int i = 10;

    }
    class Derived : Base
    {

    }

    class Marvellous
    {
        static void Main(string[] args)
        {
            // Use of GetType method
            Object ob1 = new Object();

            System.String s1 = "Marvellous Infosystems";

            Derived dobj = new Derived();

            Type type1 = ob1.GetType();

            Type type2 = s1.GetType();

            Type type3 = dobj.GetType();

            Console.WriteLine("Demonstration of GetType method");

            // Object class output
            Console.WriteLine("Information of object class");
```

```
Console.WriteLine(type1.BaseType);
Console.WriteLine(type1.Name);
Console.WriteLine(type1.FullName);
Console.WriteLine(type1.Namespace);

// string output
Console.WriteLine("Information of string class");
Console.WriteLine(type2.BaseType);
Console.WriteLine(type2.Name);
Console.WriteLine(type2.FullName);
Console.WriteLine(type2.Namespace);

// Derived output
Console.WriteLine("Information of Derived class");
Console.WriteLine(type3.BaseType);
Console.WriteLine(type3.Name);
Console.WriteLine(type3.FullName);
Console.WriteLine(type3.Namespace);

// Demonstration of equals and ReferenceEquals

Base obj1 = new Base();
Base obj2 = new Derived();
Base obj3 = new Base();
Base obj4 = obj1;

Console.WriteLine(Object.Equals(obj1, obj4));
Console.WriteLine(Object.ReferenceEquals(obj1, obj4));

string str1 = "String";
string str2 = "String";
string str3 = "String New";

Console.WriteLine("Coparision of two object");
Console.WriteLine(Object.ReferenceEquals(str1, str2));
```

```
        Console.WriteLine(Object.ReferenceEquals(obj1, obj2));
        Console.WriteLine(Object.ReferenceEquals(obj2, obj2));
        Console.WriteLine(Object.ReferenceEquals(obj1, obj3));
        Console.WriteLine(Object.ReferenceEquals(str1, str2));
        Console.WriteLine(Object.ReferenceEquals(str1, str3));

        Console.WriteLine(str1.GetHashCode());
        Console.WriteLine(str2.GetHashCode());
        Console.WriteLine(str3.GetHashCode());

        Console.WriteLine("Demonstration of toString method");
        int salary = 12000000;
        float percentage = 89.50f;

        string str4 = salary.ToString();
        string str5 = percentage.ToString();

        Console.WriteLine(str4);
        Console.WriteLine(str5);
    }
}
```