Experiment no. 4.

Title:- Implement programs on single, Multiple, Hybrid, inheritances and Interfaces.

Theory:-

Inheritance and interface are two important concepts in object-oriented programming that facilities code reuse and abstraction. In C#, inheritances allows a class to inherit properities and methods from its parent class, while interface define a set of method and properities that a class must implement.

Single inheritances:-

Single inheritances is simplest form of inheritances in which a derived class inherits the properties and methods of a single base class. In C# single inheritances is achieved using the colon symbol ":".

Syntax:-

```
class Base Class.
{
    // Properities and methods.
}
class DerivedClass : BaseClass.
{
    // Additional properties and methods.
}.
```

Multiple Inheritance:-

Multiple inheritance is a type of inheritance in which a derived class inherits properties and method from multiple base classes. However C# does not support multiple inheritance directly. Instead, it provides

It provides interface to achieve similar functionality.

Hybrid Inheritances:-

Hybrid inheritances is a type of inheritance that combines Single inheritance and Multiple inheritance. It is a complex type of inheritance that involves multiples levels of inheritances. C# does not support Hybrid inheritance directly, but it can be achieved using interfaces

Interface:-

Interface are used to define a contract for classes to implement. An interface contains only method signature, and implementation of the methods is provided by classes that implements interface. In C#, the keyword "interface" is used to define an interface.

Syntax:- e.g.

```
interface Myclass
  {  void MyMethod();  }
class Myfun : Myclass
  {  public void MyMethod()
     {  Console.WriteLine("My Class:MyMethod() is called");  }
  }
class Inf
  {  static void main(string[] args)
       {   Myfun f = new Myfun();
              f.MyMethod();
       }
  }
```

Conclusion:- In this experiment we learn how inheritance and interface are powerful concepts in C# programming that allow developers to create new classes based on existing class.