|  |
| --- |
| **Day 10 Assignment By M.Pallavi**  **04-02-2022** |

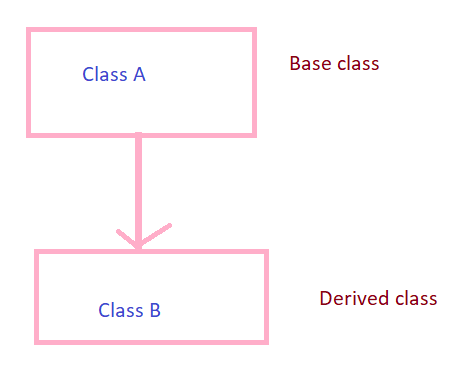
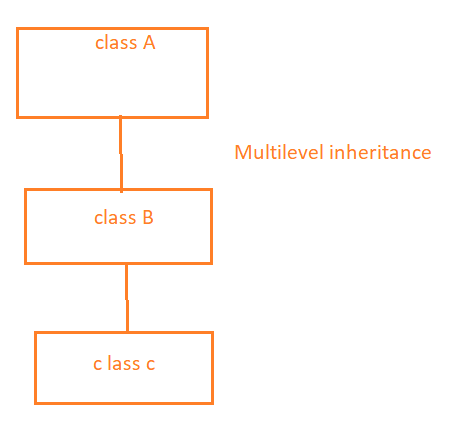
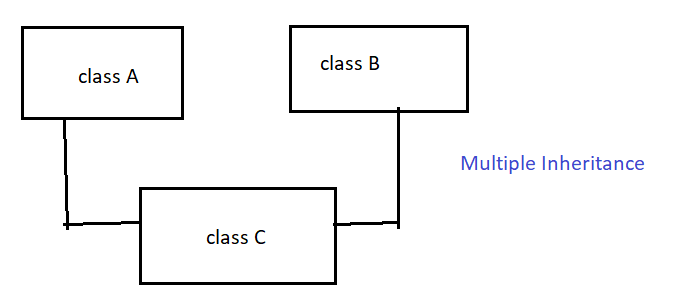
1. Write the two points discussed about inheritance in the class.

A. Inheritance is the process of Reusing base class Methods in Derived Class.

B .Inheritance Main goal is Reusability and to remove duplicate Code.

|  |
| --- |
| 2. Write example code for: a. Single inheritance, Multi level inheritance |
| Code:  using System;  namespace program\_1  {  class Animal  {  public string name="lion";  public void Display()  {  Console.WriteLine(" wild animal");  }  }  class Dog : Animal  {  public void GetName()  {  Console.WriteLine("Tiger ");  }  }  internal class Program  {  static void Main(string[] args)  {  Dog d=new Dog();    Console.WriteLine(d.name);  d.Display();  d.GetName();  }  }  } |
| Output: |

2. Pictorially represent 3 types of inheritance discussed in the class.

|  |  |
| --- | --- |
| 3.Program 1: C# program using multilevel inheritance. | |
| Code:  using System;  namespace program2  {  class GrandFather  {  public int age = 53;  public void Display()  {  Console.WriteLine("grandfather");  }  }  class Father : GrandFather  {  public int Age = 45;  public void Displayf()  {  Console.WriteLine("father ");  }  }  class son : Father  {  public int Sage = 23;  public void DisplayS()  {  Console.WriteLine("son");  }  }  internal class Program  {  static void Main(string[] args)  {  son s = new son();  s.Display();  Console.WriteLine(s.age);  s.Displayf();  Console.WriteLine(s.Age);  s.DisplayS();  Console.WriteLine(s.Sage);  Console.ReadLine();  }  }  } | |
|  | |
| 5. Why multiple inheritance is not supported for classes in C#?  The reason is that C# does not allow multiple inheritance is that multiple inheritance introduces much more complexity into a class hierarchy. To implement multiple inheritance c# uses interfaces.  In Interface looks like a class, it contains only the declaration of the members but has no implementation. It provides a contract specifying how to create an Object, without caring about the specifics of how they do the things.  An Interface is a reference type and it included only the signatures of methods, properties, events or indexers.  In order to implement an interface member, the corresponding member of the implementing class should be public, non-static, and have the same name and signature as the interface member.  6. What is polymorphism?  **Polymorphism**means providing an ability to take more than one form, polymorphism provides an ability for the [classes](https://www.tutlane.com/tutorial/csharp/csharp-classes-and-objects-with-examples) to implement different [methods](https://www.tutlane.com/tutorial/csharp/csharp-methods-functions-with-examples) called through the same name.  It also provides an ability to invoke a derived class's methods through base class reference during runtime based on our requirements. | |
| program 6: Write sample code for method overloading. |
| Code:  using System;  namespace pgmonMethodOverloading  {  class Program  {  public void Display(int a)  {  Console.WriteLine(a);  }  public void Display(int a, int b)  {  Console.WriteLine("by using overloading");  Console.WriteLine(a + b);  }  }  internal class Program1  {  static void Main(string[] args)  {  Program p = new Program();  p.Display(10);  p.Display(11, 14);  Console.ReadLine();  }  }  } |
| Output: |

|  |
| --- |
| Program: 8. Research and write sample code for method overriding using virtual ,override keyword. |
| Code:  using System;  class Animal  {  public virtual void Sound()  {  Console.WriteLine("This is parent class");  }  }  class Dog : Animal  {  public override void Sound()  {  Console.WriteLine("Dogs bark");  }  }  class Test  {  static void Main(string[] args)  {  Dog d = new Dog();  d.Sound();  }  } |
| Output: |

|  |
| --- |
| Program: 7. Write sample code for method overriding [ using new key word ] |
| Code:  using System;  public class Animal  {  public void WildAnimals()  {  Console.WriteLine(“LION”);  }  }  public class Cat : Animal  {  public new void WildAnimals()  {  Console.WriteLine(“Dog”);  }  }  public class NewProgram  {  public static void Main(string[] args)  {  Animal a = new Animal();    a.WildAnimals();  Cat c = new Cat();  c.WildAnimals();  }  } |
| Output: |