|  |
| --- |
| DAY10 ASSIGNMENT  BY  PALURU MOUNIKA  04-02-2022 |

**1.two points about inheritance.**

1.inheritance is the process of re-using the base class methode in the derived class.

2.main goal of inheritance is re-usability and remove duplicate code.

**2.example code for a. single inheritance**

**b.multi level inheritance.**

**a.single inheritance:**

|  |
| --- |
| **Program:** single inheritance |
| **Code:** using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day10\_project  {  class Algebra  {  public int add(int a, int b)  {  return a + b;  }  public int sub(int a,int b)  {  return a- b;  }  }  class TotalMaths : Algebra  {  public int mul(int a,int b)  {  return (a \* b);  }  }  internal class Program  {  static void Main(string[] args)  {  TotalMaths tm = new TotalMaths();  Console.WriteLine(tm.add(5, 6));  Console.WriteLine(tm.sub(5, 6));  Console.WriteLine(tm.mul(5, 6));  Console.ReadLine();  ; }  }  } |
| **Output:** |

**b.multi level inheritance:**

|  |
| --- |
| **Program:** multi level inheritance |
| **Code:** |
| |  | | --- | | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day10\_project  {  class Algebra  {  public int add(int a, int b)  {  return a + b;  }  public int sub(int a, int b)  {  return a - b;  }  }  class TotalMaths : Algebra  {  public int mul(int a, int b)  {  return (a \* b);  }  }  class Allsubjects : TotalMaths  {  public string methane()  {  return "CH4";  }  }  internal class Program  {  static void Main(string[] args)  {  Allsubjects obj = new Allsubjects();  Console.WriteLine(obj.add(6, 5));  Console.WriteLine(obj.sub(6, 5));  Console.WriteLine(obj.mul(5, 6));  Console.WriteLine(obj.methane());  Console.ReadLine();  ;  }  }  } | | **Output:** | |  | |

**3.Pictorial representation of 3 types of inheritance**

a. single inheritance:

**Base class**

**Deriver class**

**b.multiple inheritance:**

**Derived class**

**Base class**

**Base class**

c.multilevel inheritance:

**Base class**

Base/derived

class

**Derived class**

**4.why multiple inheritance is not supported for class in c#?**

C# compiler is designed not to support multiple inheritance because it causes ambiguityof methodes from different base class.

This cause by diamond shape problem of two clases if two clases b and c inherits from a and class d inherits from both b and c.if a methode in d calls a method defined in a,and b and c have overrriden that methode differently,then via which class deos it inherit ,b or c?.so multiple inheritance is not possible in c#.

**5.what is polymorphism ?**

- Polymorphism is ability of an object to take on many forms

- in polymorphism have 2 types

1. Overloading
2. Overriding

**6.simple code for methode over loading.**

|  |
| --- |
| **Program**: overloading |
| **Code:**   |  | | --- | | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day10\_project  {  class Algebra  {  public int add(int a, int b)  {  return a + b;  }  public int add(int a, int b,int c)  {  return a + b +c;  }  public int add(int a,int b,int c,int d)  {  return a+b+c +d;  }  }    internal class Program  {  static void Main(string[] args)  {  Algebra obj = new Algebra();  Console.WriteLine(obj.add(8, 5));  Console.WriteLine(obj.add(8, 2,6));  Console.WriteLine(obj.add(9, 1,8,4));  Console.ReadLine();  ;  }  }  } | | **Output:** | |
|  |

**7.simple program for methode overriding.**

|  |
| --- |
| **Program:** overriding |
| **Code:** |
| |  | | --- | | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Inheritance  {  class EnglishMessage  {  public void PrintHi()  {  Console.WriteLine("HI");  }  public void PrintHello()  {  Console.WriteLine("Hello");  }  public void PrintGM()  {  Console.WriteLine("Good Morning");  }  }  class TeluguMessage : EnglishMessage  {  public new void PrintGM()  {  Console.WriteLine("Subodhayam");  }  }  internal class Program  {  static void Main(string[] args)  {  TeluguMessage msg = new TeluguMessage();  msg.PrintHi();  msg.PrintHello();  msg.PrintGM();  Console.ReadLine();  }  }  } | | **Output:** | |

**8.sample code for methode overriding using virtual,override keyword.**

|  |
| --- |
| **Program:** |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Inheritance  {  class EnglishMessage  {  virtual public void PrintHi()  {  Console.WriteLine("Hi");  }  virtual public void PrintHello()  {  Console.WriteLine("Hello");  }  virtual public void PrintGM()  {  Console.WriteLine("Good Morning");  }  }  class TeluguMessage : EnglishMessage  {  override public void PrintGM()  {  Console.WriteLine("Subodhayam");  }  }  internal class Program  {  static void Main(string[] args)  {  TeluguMessage msg = new TeluguMessage();  msg.PrintHi();  msg.PrintHello();  msg.PrintGM();  Console.ReadLine();  }  }  } |
| **Output:** |
|  |