|  |
| --- |
| DAY10 ASSIGNMENT  BY  J Siva Naga Prasanna  04-02-2022 |

**1.Two points about inheritance.**

1.Inheritance is the process of reusing the base class method in the derived class.

2.Main goal of inheritance is reusability and remove duplicate code.

**2.write a code on a. Single inheritance**

**b.Multi level inheritance.**

**a.single inheritance:**

|  |
| --- |
| **Program: write a example on** single inheritancce |
| **Code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace single\_level  {  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(7, 6));  Console.WriteLine(tm.sub(6, 5));  Console.WriteLine(tm.mul(7, 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 mulilevel  {  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(8, 5));  Console.WriteLine(obj.sub(9, 5));  Console.WriteLine(obj.mul(7, 6));  Console.WriteLine(obj.methane());  Console.ReadLine();    }  }  } | | **Output:** | |  | |

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

a. single inheritance:

Super class

Sub class

**b.multiple inheritance:**

c.multilevel inheritance:

Base class

Derived class / base class

Derived class

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

C# compiler does not supports the multiple inheritance because it causes ambiguity of methods from different base class.

**5.what is polymorphism ?**

Polymorphism is ability of an object to take on many forms

There are 2 types

1. Overloading
2. Overriding

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

|  |
| --- |
| **Program**: overloading |
| **Code:**   |  | | --- | | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace overloading  {  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(7, 5));  Console.WriteLine(obj.add(7,4,2));  Console.WriteLine(obj.add(3,2,6,8));  Console.ReadLine();  ;  }  }  } | | **Output:** | |
|  |

**7.simple program for method overriding.**

|  |
| --- |
| **Program:** over riding |
| **Code:** |
| |  | | --- | | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace overriding  {  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 method over riding using virtual ,over ride keyword.**

|  |
| --- |
| **Program:** |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace virtual\_override  {  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:** |
|  |