| **DAY 10 : Morning Assignment**  **By**  **Vihar D.** |
| --- |

| **Assignment 1** |
| --- |
| **Write the points about inheritance as discussed in class.** |
| **Answer :** |
| * **Inheritance is a process of reusing the methods of the base class in derived class.** * **Inheritance is used to reduce Code Duplication as it is mandatory to have DRY code in the software industry. (DRY - Do Not Repeat)** * **The purpose of Inheritance is Reusability.** |

| **Assignment 2** |
| --- |
| **Write an example code for Single Inheritance.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace single\_inherit  {  class Algebra  {  *//Addition of 2 numbers------------------------------------------------------*  public int Add(int a, int b)  {  return a + b;  }  *//Subtraction of 2 numbers------------------------------------------------------*  public int Subt(int a, int b)  {  return a - b;  }  }  *//Inheriting TotalMath (Child Class) from Algebra (Parent Class)*  class TotalMath : Algebra  {  *//Multiplication of 2 numbers------------------------------------------------------*  public int Mult(int a, int b)  {  return a \* b;  }  }  internal class Program  {  static void Main(string[] args)  {  TotalMath math = new TotalMath();  Console.WriteLine("\n Single Inheritance ------------------------------------------------------\n");  Console.WriteLine("\n Sum of 9 and 6 is {0}", math.Add(9, 6));  Console.WriteLine("\n Difference of 9 and 6 is {0}", math.Subt(9, 6));  Console.WriteLine("\n Product of 9 and 6 is {0}", math.Mult(9, 6));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 3** |
| --- |
| **Write an example code for Multi-level Inheritance.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace multi\_inherit  {  class Algebra  {  *//Addition of 2 numbers------------------------------------------------------*  public int Add(int a, int b)  {  return a + b;  }  *//Subtraction of 2 numbers------------------------------------------------------*  public int Subt(int a, int b)  {  return a - b;  }  }  *//Inheriting TotalMath (Child Class) from Algebra (Parent Class)*  class TotalMath : Algebra  {  *//Multiplication of 2 numbers------------------------------------------------------*  public int Mult(int a, int b)  {  return a \* b;  }  }  *//Inheriting TotalMath (Child Class) from Algebra (Parent Class)*  class OverallMath : TotalMath  {  *//Division of 2 numbers------------------------------------------------------*  public int Div(int a, int b)  {  return a / b;  }  *//Modulus of 2 numbers------------------------------------------------------*  public int Mod(int a, int b)  {  return a % b;  }  }  internal class Program  {  static void Main(string[] args)  {  OverallMath math = new OverallMath();  Console.WriteLine("\n Multi-Level Inheritance ------------------------------------------------------\n");  Console.WriteLine("\n Sum of 9 and 6 is {0}", math.Add(9, 6));  Console.WriteLine("\n Difference of 9 and 6 is {0}", math.Subt(9, 6));  Console.WriteLine("\n Product of 9 and 6 is {0}", math.Mult(9, 6));  Console.WriteLine("\n Division of 15 and 3 is {0}", math.Div(15, 3));  Console.WriteLine("\n Modulus of 9 and 6 is {0}", math.Mod(9, 6));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 4** |
| --- |
| **Pictorially represent 3 types of inheritance.** |
| **Answer :** |
| **Single Inheritance Multilevel Inheritance Multiple Inheritance** |

| **Assignment 5** |
| --- |
| **Why is Multiple Inheritance not supported for C# classes ?** |
| **Answer :** |
| **C# does not support multiple inheritance because of the diamond problem which is associated with multiple class inheritance.**    **For Instance,**   * **Here, classes B & C are inherited from class A . Then, another class D is inherited from classes B & C.** * **If class D calls a method from class A and class D has not overridden the invoked method. But, class B & C has already been overridden.** * **The issue occurring here is called an ambiguity problem which occurs while invoking the methods.** * **Hence, Multiple class inheritance is not supported on C#.** |

| **Assignment 6** |
| --- |
| **What is Polymorphism ?** |
| **Answer :** |
| **Polymorphism means ‘ many forms ’ ; it occurs when many classes are related to each other by inheritance . It basically means one object can have many forms. Poly means “many” and morph means “alter”.**  **Definition : Polymorphism is the ability of a class to have the same name but multiple implementations. It's one of the main principles of OOP. It's an ability of an object to take on many forms.**  **Live Example : one man many personas.**    **Diagram :**    **There are 2 types of Polymorphism :**   1. **Method Overloading 2. Method Overriding** |

| **Assignment 7** |
| --- |
| **Write an example code for Method Overloading.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace method\_overload  {  class Maths  {  *//Adding 2 numbers-------------------------*  public int Add(int a, int b)  {  return a + b;  }  *//Adding 3 numbers-------------------------*  public int Add(int a, int b , int c)  {  return a + b + c;  }  }  internal class Program  {  static void Main(string[] args)  {  Maths obj = new Maths();  Console.WriteLine("\n Method Overloading ----------------");  Console.WriteLine("Addition of 5 and 10 is :{0}", obj.Add(5, 10));  Console.WriteLine("Addition of 5, 10 & 15 is :{0}", obj.Add(5, 10, 15));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 8** |
| --- |
| **Write an example code for Method Overriding using the new keyword.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace method\_overload  {  class Maths  {  *//Adding 2 numbers-------------------------*  public int Add(int a, int b)  {  return a + b;  }  *//Adding 3 numbers-------------------------*  public int Add(int a, int b , int c)  {  return a + b + c;  }  }  internal class Program  {  static void Main(string[] args)  {  Maths obj = new Maths();  Console.WriteLine("\n Method Overloading ----------------");  Console.WriteLine("Addition of 5 and 10 is :{0}", obj.Add(5, 10));  Console.WriteLine("Addition of 5, 10 & 15 is :{0}", obj.Add(5, 10, 15));  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 9** |
| --- |
| **Write an example code for Method Overriding using the new keyword.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace method\_override\_new  {  class EnglishGreet  {  *//Print Hi-----------------------*  public void PrintHi()  {  Console.WriteLine("Hi");  }  *//Print Hello-----------------------*  public void PrintHello()  {  Console.WriteLine("Hello");  }  *//Print Good Morning-----------------------*  public void PrintGM()  {  Console.WriteLine("Good Morning");  }  }  *//Print Subhodhayam-----------------------*  class TeluguGreet : EnglishGreet  {  public new void PrintGM()  {  Console.WriteLine("Subhodhayam");  }  }  internal class Program  {  static void Main(string[] args)  {  TeluguGreet obj = new TeluguGreet();  Console.WriteLine("\n Method Overriding using new keyword-----------");  obj.PrintHi();  obj.PrintHello();  obj.PrintGM();  Console.ReadLine();  }  }  } |
| **Output :** |
|  |

| **Assignment 10** |
| --- |
| **Write an example code for Method Overriding using virtual and override keywords.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace method\_override\_virtual  {  class EnglishGreet  {  *//Print Hi-----------------------*  public void PrintHi()  {  Console.WriteLine("Hi");  }  *//Print Hello-----------------------*  public void PrintHello()  {  Console.WriteLine("Hello");  }  *//Print Good Morning-----------------------*  public virtual void PrintGM()  {  Console.WriteLine("Good Morning");  }  }  *//Print Subhodhayam-----------------------*  class TeluguGreet : EnglishGreet  {  public override void PrintGM()  {  Console.WriteLine("Subhodhayam");  }  }  internal class Program  {  static void Main(string[] args)  {  TeluguGreet obj = new TeluguGreet();  EnglishGreet obj2 = new EnglishGreet();  Console.WriteLine("\n Method Overriding using " +  "virtual and override keywords-----------");  obj.PrintHi();  obj.PrintHello();  obj2.PrintGM();  obj.PrintGM();  Console.ReadLine();  }  }  } |
| **Output :** |
|  |