|  |
| --- |
| **Day -10 Morning Assignments**  **By**  **Manoj Yekolla**  **04-Feb-2022** |

|  |
| --- |
| **1. Write the two points discussed about inheritance in the class ?** |
| * Inheritance is the process of re-using base class methods in the derived class. * Inheritance main goal is Re-usability and to remove duplicate code. |

|  |
| --- |
| **2(a). Write example code for-Single inheritance ?** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day10\_Project1  {  class Algebra  {  public int Add(int a, int b)  {  return a + b;  }  public int Sub(int a, int b)  {  return a - b;  }  }  class Math : Algebra  {  public int Mul(int a, int b)  {  return a \* b;  }  }    internal class Program  {    static void Main(string[] args)  {  Math obj= new Math();  Console.WriteLine("\nsum 11 and 2 is {0}" ,obj.Add(11,2));  Console.WriteLine("\nsub 11 and 2 is {0}", obj.Sub(11, 2));  Console.WriteLine("\nmul 11 and 2 is {0}", obj.Mul(11, 2));  Console.ReadLine();    }  }  } |
| Output :  Screenshot (160) |

|  |
| --- |
| **2(b). Multi level inheritance ?** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day10\_Project2  {  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 OverAllMaths : TotalMaths  {    public int Div(int a, int b)  {  return a / b;  }      }  internal class Program  {  static void Main(string[] args)  {  OverAllMaths obj = new OverAllMaths();    Console.WriteLine("\n Sum of Two Numbers 8 and 9 is {0}", obj.Add(8, 9));  Console.WriteLine("\n Difference of Two Numbers 8 and 9 is {0}", obj.Sub(8, 9));  Console.WriteLine("\n Product of Two Numbers 8 and 9 is {0}", obj.Mul(8, 9));  Console.WriteLine("\n Division of Two Numbers 8 and 9 is {0}", obj.Div(8, 9));    Console.ReadLine();  }  }  } |
| OutPut :  Screenshot (162) |

|  |
| --- |
| **3. Pictorially represent 3 types of inheritance discussed in the class ?** |
| 1. **Single Level Inheritance**   Single Inheritance |
| 1. **Multilevel Inheritance**   **multi** |
| 1. **Multiple Inheritance**   **download** |

|  |
| --- |
| **4. Why multiple inheritance is not supported for classes in**  **C# ?** |
| C# compiler is designed not to support multiple inheritence **because it causes ambiguity of methods from different base class**.  This is Cause by diamond Shape problems of two classes If two classes B and C inherit from A, and class D inherits from both B and C.  So., multiple inheritance is not possible in C#. |

|  |
| --- |
| **5. What is polymorphism ?** |
| Ans :  Polymorphism is a Greek word, meaning "one name many forms". In other words, one object has many forms or has one name with multiple functionalities. Polymorphism **provides the ability to a class to have multiple implementations with the same name**.  Eg :  For example, think of a base class called Animal that has a method called animalSound(). Derived classes of Animals could be Pigs, Cats, Dogs, Birds - And they also have their own implementation of an animal sound (the pig oinks, and the cat meows, etc.): |
| Polymorphism_In_Csharp |

|  |
| --- |
| **6. Write sample code for method overloading ?** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day10\_Project3  {  class Mathematics  {  public int Add(int a , int b)  {  return a + b;  }  public int Add(int a, int b,int c)  {  return a + b+c;  }  }  internal class Program  {  static void Main(string[] args)  {  Mathematics obj = new Mathematics();  Console.WriteLine(" Add of Two Numbers 8 and 10 is : {0}", obj.Add(8, 10));  Console.WriteLine(" Add of Three Numbers 8 ,12and 10 is : {0} ", obj.Add(8,12, 10));  //Console.WriteLine(obj.Add(2, 8));  Console.ReadLine();  }  }  } |
| Output :  Screenshot (165) |

|  |
| --- |
| **7. Write sample code for method overriding [ using new key word ]** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day10\_Project4  {  class EnglishMsg  {  public void PrintHi()  {  Console.WriteLine("Hi");  }  public void PrintHlo()  {  Console.WriteLine("Hlo");  }  public void PrintGm()  {  Console.WriteLine("Gm");  }  }  class TeluguMsg :EnglishMsg  {  public new void PrintGm()  {  Console.WriteLine("subhodhyam");  }  }  internal class Program  {  static void Main(string[] args)  {  TeluguMsg obj = new TeluguMsg();  obj.PrintHi();  obj.PrintHlo();  obj.PrintGm();    Console.ReadLine();  }  }  } |
| Output :  Screenshot (167) |

|  |
| --- |
| **8. write sample code for method overriding using virual, override keyword. ?** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day10\_Project5  {  class English  {  public void Hi()  {  Console.WriteLine("Hi");  }  public void Hlo()  {  Console.WriteLine("Hlo");  }  public virtual void Gm()  {  Console.WriteLine("Gm");  }  }  class Telugu : English  {  public override void Gm()  {  Console.WriteLine("Subhodhyam");  }  }  internal class Program  {  static void Main(string[] args)  {  Telugu obj = new Telugu();  obj.Hi();  obj.Hlo();  obj.Gm();    Console.ReadLine();  }  }  } |
| Output :  Screenshot (169) |