|  |
| --- |
| Feb 4th Evening Assignment  By Surya Teja Chandolu |

|  |
| --- |
| 1. Research and try to understand what is Abstraction |
| Abstraction is the process of hiding certain details and showing only essential information to the user.  Abstraction can be achieved with either abstract classes or interfaces  Abstract class is a restricted class that cannot be used to create objects |

|  |
| --- |
| 1. Write the 2 main uses of Abstract class by using the example discussed in the class. |
| * Abstract is used for code reusability. * Abstract class is also used to enforce the child class to use the abstract method In the child class for compulsory. |
| Code: |
| using System;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author: Surya Teja  \* Purpose: Salary Process  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace SalaryProcess  {  abstract class Salary  {  public int GetPF(int basic)  {  return 12 \* basic / 100;  }  public int GetHRA(int basic)  {  return 40 \* basic / 100;  }  public abstract int GetCA();  public abstract int GetSA();  }  class Microsoft : Salary  {  public override int GetCA()  {  return 6000;  }  public override int GetSA()  {  return 5000;  }  }  class Google : Salary  {  public override int GetCA()  {  return 10000;  }  public override int GetSA()  {  return 9000;  }  }  class Tesla : Salary  {  public override int GetCA()  {  return 8000;  }  public override int GetSA()  {  return 6000;  }  }  class Facebook : Salary  {  public override int GetCA()  {  return 7000;  }  public override int GetSA()  {  return 5000;  }  }  internal class Program  {  static void Main(string[] args)  {  Console.WriteLine("Process Completed");  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. Create one more example of your choice to demonstrate abstract class |
| Code: |
| using System;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author: Surya Teja  \* Purpose: Car Design  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace CarDesign  {  abstract class Car  {  public bool Engine()  {  return true;  }  public abstract double Length();  public abstract double Width();  public abstract int MaxSpeed();  }  class Audi : Car  {  public override double Length()  {  return 17.4;  }  public override double Width()  {  return 6.38;  }  public override int MaxSpeed()  {  return 250;  }  }  class BMW : Car  {  public override double Length()  {  return 17.12;  }  public override double Width()  {  return 6.24;  }  public override int MaxSpeed()  {  return 250;  }  }  internal class Program  {  static void Main(string[] args)  {  Console.WriteLine("Completed");  Console.ReadLine();  }  }  } |
| Output: |
|  |