|  |
| --- |
| Day10 Evening Assignment  By  Anusha Bellala |

|  |
| --- |
| 1. Research and try to understand what is Abstraction. |
| Data 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. |

|  |
| --- |
| Write the 2 main uses of Abstract class  by using the example discussed in the class. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp1  {  abstract class Salary  {  public int GetPF(int basic)  {  return 12 \* basic / 100;  }  public int HRA(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 7000;  }  }  class Google : Salary  {  public override int GetCA()  {  return 10000;  }  public override int GetSA()  {  return 10000;  }  }  class IBM : Salary  {  public override int GetCA()  {  return 4000;  }  public override int GetSA()  {  return 6000;  }  }  class Facebook : Salary  {  public override int GetCA()  {  return 20000;  }  public override int GetSA()  {  return 10000;  }  }  internal class Program  {  static void Main(string[] args)  {  //Microsoft  //Google  //IBM  //Facebook  Console.WriteLine("Complete Processing");  Console.ReadLine();  }  }  } |
| Ouput: |

Uses of Abstraction class:

|  |
| --- |
| 1.code reusability. |
| 2.To enforce the abstract methods must be overridden in the derived class. |

|  |
| --- |
| 3.Example of Abstract class of abstract method draw().its implement ion is provided in derived classes: Rectangle, Circle. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp2  {  public abstract class Shape  {  public abstract void draw();  }  public class Rectangle : Shape  {  public override void draw()  {  Console.WriteLine("drawing rectangle...");  }  }  public class Circle : Shape  {  public override void draw()  {  Console.WriteLine("drawing circle...");  }  }  internal class Program  {  static void Main(string[] args)  {  Shape s;  s = new Rectangle();  s.draw();  s = new Circle();  s.draw();  Console.ReadLine();  }  }  } |
| Ouput: |