|  |
| --- |
| Abstraction Assignment  By  Nanam Vaishnavi  04 – Feb - 2022 |

|  |
| --- |
| 1. **Research and try to understand what is Abstraction** |
| Ans:  Abstraction refers to hide implementation or background details and it will display only essential information about the data to the outside world.   1. The abstract methods in the abstract class are implemented actually in the derived classes. 2. No object can be created of the abstract class i.e., it can’t be instantiated. 3. Abstract class may define abstract methods. 4. Derived classes of the abstract class must implement all abstract methods.   **Shapes**  **Square**  **Circle**  **Rectangle** |

|  |
| --- |
| 1. **Write the 2 main uses of Abstract class by using the example.** |
| Ans :   1. Code Re-Usabaility. 2. Enforcing the derived class to override the abstract class.   Example :  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // Author : Nanam Vaishnavi  // Purpose : Abstract class  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Abstraction  {  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 30000;  }  public override int GetSA()  {  return 20000;  }  }  class Google : Salary  {  public override int GetCA()  {  return 10000;  }  public override int GetSA()  {  return 40000;  }  }  class IBM : Salary  {  public override int GetCA()  {  return 50000;  }  public override int GetSA()  {  return 30000;  }  }  class Facebook : Salary  {  public override int GetCA()  {  return 5000;  }  public override int GetSA()  {  return 3000;  }  }  internal class Program  {  static void Main(String[] args)  {  // Microsoft  // Google  // IBM  //Facebook  Console.WriteLine("Completed Processing");  Console.ReadLine();  }  }  } |

|  |
| --- |
| **3)Create one more example of your choice to demonstrate abstract class** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Abstraction  {  abstract class Shape  {  public abstract double Area();  }    class Square : Shape  {  private double side = 0;  public Square(double s)  {  side = s;  }  public override double Area()  {  return (side \* side);  }  }    class Circle : Shape  {  private double radius;  public Circle(double r)  {  radius = r;  }  public override double Area()  {  return 3.14\*radius\*radius;  }  }  internal class Program  {  static void Main(string[] args)  {  Square s = new Square(8);  Console.WriteLine(s.Area());  Circle c = new Circle(6.5);  Console.WriteLine(c.Area());    Console.ReadLine();  }  }  } |