**Smt. Chandaben Mohanbhai Patel Institute of Computer Applications**

**Bachelors of Computer Applications**

**Semester – V**

**CA 315 – Advanced Programming with .NET Framework**

**Practical Assignment – II**

**ID – 18BCA026 Name- Saurabh Kumar V Mishra**

|  |  |  |
| --- | --- | --- |
| 1 | * Create a new project, and include in it the class Person that you just created. * Create a class "Student" and another class "Teacher", both descendants of "Person". * The class "Student" will have a public method "GoToClasses", which will write on screen "I’m going to class." * The class "Teacher" will have a public method "Explain", which will show on screen "Explanation begins". Also, it will have a private attribute "subject", a string. * The class Person must have a method "SetAge (int n)" which will indicate the value of their age (eg, 20 years old). * The student will have a public method "ShowAge" which will write on the screen "My age is: 20 years old" (or the corresponding number). * You must create another test class called "StudentAndTeacherTest" that will contain "Main" and:   + Create a Person and make it say hello   + Create a student, set his age to 21, tell him to Greet and display his age   + Create a teacher, 30 years old, ask him to say hello and then explain.   **Code:-**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_1  {  class person  {  public void hello()  {  Console.WriteLine("Hello");  }  public void SetAge(int n)  {  Console.WriteLine(n + " years old");  }  }  class student : person  {  public void GoToClasses()  {  Console.WriteLine("I’m going to class.");  }  }  class teacher : person  {  String subject;  public void Explain()  {  Console.Write("Enter subject:");  subject = Console.ReadLine();  Console.WriteLine("Explanation begins " + subject);  }  }  class StudentAndTeacherTest  {  static void Main(string[] args)  {  Console.WriteLine("Object of Person");  person p1 = new person();  p1.hello();  Console.WriteLine("Object of student\n");  student sobj = new student();  sobj.SetAge(21);  sobj.GoToClasses();  Console.WriteLine("Object of teacher\n");  teacher tobj = new teacher();  tobj.SetAge(30);  tobj.hello();  tobj.Explain();  Console.Read();  }  }  }  **Output-:** | |
| 2 | Create a class "PhotoAlbum" with a private attribute "numberOfPages."  It should also have a public method "GetNumberOfPages", which will return the number of pages.  The default constructor will create an album with 16 pages. There will be an additional constructor, with which we can specify the number of pages we want in the album.  Create a class "BigPhotoAlbum" whose constructor will create an album with 64 pages.  Create a test class "AlbumTest" to create an album with its default constructor, one with 24 pages, a "BigPhotoAlbum" and show the number of pages that the three albums have.  **Code-:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_2  {  class PhotoAlbum  {  protected int numberOfPages;  public PhotoAlbum()  {  numberOfPages = 16;  }  public int GetNumberOfPages()  {  return numberOfPages;  }  }  class AlbumTest : PhotoAlbum  {    public AlbumTest()  {  numberOfPages = 24;  }    }  class BigPhotoAlbum : PhotoAlbum  {    public BigPhotoAlbum()  {  numberOfPages = 64;  }  }  class Program  {  static void Main(string[] args)  {  PhotoAlbum PA = new PhotoAlbum();  Console.WriteLine("Number of pages: " + PA.GetNumberOfPages());  AlbumTest Ah = new AlbumTest();  Console.WriteLine("Number of pages: " + Ah.GetNumberOfPages());  BigPhotoAlbum bh = new BigPhotoAlbum();  Console.WriteLine("Number of pages: " + bh.GetNumberOfPages());  Console.Read();  }  }  }  **Output-:** | |
| 3 | Create a project and the corresponding classes for the following class diagram:  https://3.bp.blogspot.com/-LaghNM7b-2w/XK9bhHNmJhI/AAAAAAAAGIo/JbeVHU60gVs45mMJmLEROsMI6UscZvSRQCLcBGAs/s1600/diagram-shapes1.jpeg  Code:-  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_3  {  class shap  {  protected double radius, x, y, side1, side2;  public shap()  {  side1 = 2;  side2 = 3;  y = 1;  x = 2;  radius = 5;  }  public void lication()  {  Console.WriteLine("Location is : " + x + " , " + y);  }  public String Tostring()  {  return "This is String";  }  public double area()  {  return (radius \* 3.14);  }  public double rectangle()  {  return (side1 \* side2);  }  }  class rectangle : shap  {  //protected double side1, side2;  public rectangle()  {  side1 = 56;  side2 = 34;  }  }    class circle : shap  {  //protected double radius;  public circle()  {  radius = 12;  }  }  class Location : shap  {  public Location()  {  x = 3;  y = 5;  }  }  class Program  {  static void Main(string[] args)  {  rectangle ra = new rectangle();  Console.WriteLine("Parameter of rectangle: "+ra.rectangle());  circle ca = new circle();  Console.WriteLine("Area of circle: " + ra.area());  Location la = new Location();  la.lication();  Console.WriteLine( la.Tostring());  Console.Read();  }  }  }  Output-: | |
| 4 | Create a project and the corresponding classes refers to class diagram.  Each class must include the attributes and methods shown in the diagram, as well as Get and Set methods for Vehicle and "Has" methods ("HasDualSlidingDoors") for MiniVan.  You must create also a test program, which will create an object belonging to each class and tell it to "Drive".  D:\One Drive\OneDrive - Charotar University\CA 315 ADNF\Class Diagram For Assignment.png  Code-:  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_4  {  class vehical  {  protected String make;  protected String model;  protected String year;  public vehical()  {  //make = "Honda";  //model = "BS4";  //year = "2015";  }  public vehical(String make, String model, String year)  {  this.make = make;  this.model = model;  this.year = year;  }  public void Accelerate()  {  Console.WriteLine("Car name:" + make);  Console.WriteLine("Car model:" + model);  Console.WriteLine("Car year:" + year);  }  public void Decelerate()  {  Console.WriteLine("Car name:" + make);  Console.WriteLine("Car model:" + model);  Console.WriteLine("Car year:" + year);  }  public void Drive()  {  Console.WriteLine(make + " is Drive");    }  public void Start()  {  Console.WriteLine(make + " is Start");  }  public void Stop()  {  Console.WriteLine(make + " is stop");  }  }  class car: vehical  {  protected String carname;  public car()  {  make = "BMW";    }  }  class sportscar: car  {  //protected String carname;  public sportscar()  {  make = "Sportcar";  }  }  class van: vehical  {  //protected String vanname;  public van()  {  make = "maruti";  }  }  class ExcursionVan : van  {  public ExcursionVan()  {  make = "Honda";  }  }  class minivan : van  {  protected bool cargo\_net;  protected bool dual\_siding\_doors;  public minivan()  {  cargo\_net = true;  dual\_siding\_doors = false;  }  public minivan(bool cargo\_net, bool dual\_siding\_doors)  {  this.cargo\_net = cargo\_net;  this.dual\_siding\_doors = dual\_siding\_doors;  }  public void SetDualSlidingDoor(bool dual\_siding\_doors)  {  this.dual\_siding\_doors = dual\_siding\_doors;  }  public void setcargoNet(bool cargo\_net)  {  this.cargo\_net = cargo\_net;  }  public bool HasCargoNet()  {  return cargo\_net;  }  public bool HasDualSlidingDoor()  {  return dual\_siding\_doors;  }  }  class TestVehical  {  static void Main(string[] args)  {  car myCar = new car();  myCar.Drive();  sportscar mySportsCar = new sportscar();  mySportsCar.Accelerate();  van myVan = new van();  myVan.Stop();  minivan myMiniVan = new minivan();  myMiniVan.Start();  ExcursionVan myExcursionVan = new ExcursionVan();  myExcursionVan.Drive();  Console.Read();  }  }  }  Output:- | |
| 5 | Create the project named "Shapes" (january 8th), adding a class named "Square" to it. For each object of square, we will store its starting X and Y coordinates (the upper left corner, already stored as a "Location") and the length of its side.  You will have to create:  - A suitable constructor, to assign starting values to X, Y and the side.  - A Move method, to change X and Y coordinates.  - A Scale method, to change its side (for example, a scale factor of 2 would turn a side of 3 into 6).  - A method ToString, to return a string with its data (for example: "Corner (10,5), side 7".  - Define "GetArea", to find area of square.  Code:-  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_5  {  class square  {  int x, y;  int side;  public square(int x, int y, int side)  {  this.x = x;  this.y = y;  this.side = side;  }  public void move()  {  Console.WriteLine("How many side do you want to move square");  String n = Console.ReadLine();  x = x + Convert.ToInt32(n);  y = y + Convert.ToInt32(n);  }  public string ToString()  {  return "Corner (" + x + "," + y + ")" + "side =" + side;  }  public void scale()  {  int sf = side \* 2;  Console.WriteLine($"The scale factor of 2 turns {sf} to {side}.");  }  public void Area()  {  Console.WriteLine("Area of square:" + (side \* side));  }  }  class Program  {  static void Main(string[] args)  {  square s = new square(5, 3, 3);  s.move();  s.scale();  string s1 = s.ToString();  Console.WriteLine(s1);  s.Area();  Console.Read();    }  }  }  Output-: | |
| 6 | create a project and the corresponding classes for this classes diagram.  Each class must include the attributes and methods shown in the diagram. Consider that all cardinalities are 1:1  D:\One Drive\OneDrive - Charotar University\CA 315 ADNF\Class Diagram For Assignment 2.png  Code-:  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_6  {  class item : OrderDetail  {  protected string weight;  protected string description;  public item()  {  weight = "2kg";  description = "Computer Disk";  }  public double getPriceForQuantity()  {  return 6 \* 70;  }  public double getWeight()  {  return 5;  }  public string GetShippingWeight()  {  return weight;  }  public string GetDescription()  {  return description;  }  public void SetShippingWeight(string shippingWeight)  {  this.weight = shippingWeight;  }  public void SetDescription(string description)  {  this.description = description;  }  public void show()  {  Console.WriteLine($"Item Name: {description} and weight: {weight}");  }  }  class Order  {  protected DateTime date;  protected string status;  protected OrderDetail[] o;  public Order()  {  status = "Baroda";  }  public double calcTax()  {  return 0;  }  public double calcTotal()  {  return 0;  }  public double calcTotalWeight()  {  return 0;  }  public DateTime GetDate()  {  return date;  }  public string GetStatus()  {  return status;  }  public void SetDate(DateTime date)  {  this.date = date;  }  public void SetStatus(string status)  {  this.status = status;  }  public void show()  {  Console.WriteLine("Order status is: " + status);  }  }  class OrderDetail : Order  {  protected item[] i;  protected double tax, quantity;  public OrderDetail()  {  quantity = 0.0;  tax = 0.0;  }  public double caltotal()  {  return 0;  }  public double calTotalWeight()  {  return 0;  }  public DateTime getdate()  {  return date;  }  public string getstatus()  {  return status;  }  public void setdate(DateTime date)  {  this.date = date;  }  public void setstatus(string status)  {  this.status = status;  }  }  class customer  {  protected string name, address;  protected OrderDetail[] o;  public customer()  {  name = "Mohan Das";  address = "Mohani cornar,Baroda";  }  public void show()  {  Console.WriteLine($"Customer name: {name} and Address : {address}");  }  }  class Program  {  static void Main(string[] args)  {  double total;  customer c =new customer();  c.show();  item i = new item();  i.show();  total = i.getPriceForQuantity();  Console.WriteLine($"The Price For Total Quantity Is :{total}");  Order o = new Order();  o.show();  Console.Read();  }  }  }  Output-: | |
| 7 | Expand program 5 (shapes + square), so that it can also store data about "Colored Circles".  Please note the following:   * The "color" must be an integer, not type "Color" * You must create two constructors, as shown * You must also create methods ToString, Scale, Move * Instead of "String getColors", create "int GetColor()" * Define GetArea() * "Main()" must not be part of "ColoredCircle", but of the test program.   Code:-  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_7  {  class square  {  protected int x, y;  protected int side;  public square(int x, int y, int side)  {  this.x = x;  this.y = y;  this.side = side;  }  public void move()  {  Console.WriteLine("How many side do you want to move square");  String n = Console.ReadLine();  x = x + Convert.ToInt32(n);  y = y + Convert.ToInt32(n);  }  public string ToString()  {  return "Corner (" + x + "," + y + ")" + "side =" + side;  }  public void scale()  {  int sf = side \* 2;  Console.WriteLine($"The scale factor of 2 turns {sf} to {side}.");  }  public void Area()  {  Console.WriteLine("Area of square:" + (side \* side));  }  }  class shapes  {  protected int color;  public shapes(int color)  {  this.color = color;  }  public string ToString()  {  return "Corner ("+color+")";  }  public string getColor()  {  Console.WriteLine("Enter color:");  String color = Console.ReadLine();  return color;  }  public int Getcolor(String color)  {  return this.color = Convert.ToInt32(color);  }  public void Getarea()  {  Console.WriteLine("Enter radius:");  int r = Convert.ToInt32(Console.ReadLine());  double pi = 3.14;  Console.WriteLine("Area of CircleShape: " + (pi \* r \* r));  Console.WriteLine("Color of CircleShape: " + color);  }  }  class Program  {  static void Main(string[] args)  {  Console.WriteLine("--------Square--------");  square s = new square(5, 3, 3);  s.move();  s.scale();  string s1 = s.ToString();  Console.WriteLine(s1);  s.Area();  Console.WriteLine("--------Circle--------");  shapes sp = new shapes(2);  sp.Getcolor(sp.getColor());  string s2 = sp.ToString();  Console.WriteLine(s2);  sp.Getarea();  Console.Read();  }  }  }  Output:- | |
| 8 | Create a class named "Table". It must have a constructor, indicating the width and height of the board. It will have a method "ShowData" which will write on the screen the width and that height of the table. Create an array containing 10 tables, with random sizes between 50 and 200 cm, and display all the data.  Code:-  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_8  {  class Table  {  int width;  int height;  public Table()  {  width = 56;  height = 45;  }  public Table(int width,int height)  {  this.width = width;  this.height = height;  }  public void ShowData()  {  Console.WriteLine("Width of table: " + width);  Console.WriteLine("Height of table: " + height);  }  }  class Program  {  static void Main(string[] args)  {  int[] width = new int[10] { 78, 98, 65, 90, 100, 120, 111, 56, 78, 199 };  int[] height = new int[10] { 112, 115, 123, 156, 145, 178, 113, 110, 134, 195 };    for(int i = 1; i <= width.Length; i++)  {  Table wh = new Table(width[i], height[i]);  wh.ShowData();  }  Console.Read();  }  }  }  Output-: | |
| 9 | Create a class "House", with an attribute "area", a constructor that sets its value and a method "ShowData" to display "I am a house, my area is 200 m2" (instead of 200, it will show the real surface). Include getters an setters for the area, too.  The "House" will contain a door. Each door will have an attribute "color" (a string), and a method "ShowData" wich will display "I am a door, my color is brown" (or whatever color it really is). Include a getter and a setter. Also, create a "GetDoor" in the house.  A "SmallApartment" is a subclass of House, with a preset area of 50 m2.  Also create a class Person, with a name (string). Each person will have a house. The method "ShowData" for a person will display his/her name, show the data of his/her house and the data of the door of that house.  Write a Main to create a SmallApartment, a person to live in it, and to show the data of the person.  Code:-  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_9  {  class House  {  protected int area;  protected Door door;  public House()  {  area = 200;  }  public House(int area)  {  this.area = area;  door = new Door();  }  public int Area  {  get { return area; }  set { area = value; }  }  public Door Door  {  get { return door; }  set { door = value; }  }  public virtual void ShowData()  {  Console.WriteLine("I am a house, my area is {0} m2.", area);  }  }  class Door  {  protected String color;  public Door()  {  color = "Brown";  }  public Door(String color)  {  this.color = color;  }  public string Color  {  get { return color; }  set { color = value; }  }  public void ShowData()  {  Console.WriteLine("I am a door, my color is {0}.", color);  }  }  class SmallApartment : House  {  public SmallApartment()  : base(50)  {    }  public override void ShowData()  {  Console.WriteLine("I am an apartment, my area is " + area + " m2");  }  }  class Person  {  protected String name;  protected House home;  public Person()  {  name = "Rohan";  home = new House(100);  }  public Person(String name, House home)  {  this.name = name;  this.home = home;  }  public string Name  {  get { return name; }  set { name = value; }  }  public House House  {  get { return home; }  set { home = value; }  }  public void ShowData()  {  Console.WriteLine("My name is {0}.", name);  home.ShowData();  home.Door.ShowData();  }  }  class Program  {  static void Main(string[] args)  {  SmallApartment sa = new SmallApartment();  Person p = new Person();  p.House = sa;  p.ShowData();  Console.Read();  }  }  }  Output-: | |
| 10 | | Refer to exercise no. 8, create a project named "Tables2", based on the "Tables" project.  In it, create a class "CoffeeTable" that inherits from "Table". Its method “ShowData", besides writing the width and height, must display "(Coffee table)."  Create an array that contains 5 tables and 5 coffee tables. The tables must have random sizes between 50 and 200 cm, and the coffee tables from 40 to 120 cm. Show all their data.  Code-  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_10  {  class Table  {  protected double width, height;  public Table()  {  width = 0;  height = 0;  }  public Table(double width,double height)  {  this.width = width;  this.height = height;  }  public double Width  {  get { return width; }  set { width = value; }  }  public double Height  {  get { return height; }  set { height = value; }  }  public virtual void ShowData()  {  Console.WriteLine($"Table Width:{width} and Height:{height}.");  }  }  class CoffeeTable : Table  {  public CoffeeTable(double width, double height)  {  this.width = width;  this.height = height;  }  public override void ShowData()  {  Console.WriteLine($"CoffeeTable Width:{width} and Height:{height}.");  }  }  class Program  {  static void Main(string[] args)  {  Table[] table = new Table[5];  Random rmd = new Random();  for (int i = 1; i <= 5; i++)  {  if ((i % 2 == 0) && (i!=1))  {  table[i - 1] = new Table(rmd.Next(50, 200), rmd.Next(50, 200));  table[i - 1].ShowData();  }  else  {  table[i - 1] = new CoffeeTable(rmd.Next(40, 120), rmd.Next(40, 120));  table[i - 1].ShowData();  }  }  Console.Read();  }  }  }  Output-: |
| 11 | | Create a class "Encrypter" to encrypt and decrypt text.  It will have a "Encrypt" method, which will receive a string and return another string. It will be a static method, so that we do not need to create any object of type "Encrypter".  There will be also a "Decrypt" method.  In this first approach, the encryption method will be a very simple one: to encrypt we will add 1 to each character, so that "CMPICA" would become "DNQJDB", and to decrypt we would subtract 1 to each character.  An example of use might be  string newText = Encrypter.Encrypt("CMPICA");  Code:-  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_11  {  class Encrypter  {  public static string Encrypt(String text)  {  int wordInt = 0;  char word = ' ';  string wordEncrypt = " ";  for(int i=0;i<text.Length;i++)  {  wordInt = (int)text[i] + 1;  word = (char)wordInt;  wordEncrypt += word.ToString();  }  return wordEncrypt;  }  public static string Decrypt(string text)  {  int wordInt = 0;  char word = ' ';  string wordDecript = "";  for (int i = 0; i < text.Length; i++)  {  wordInt = (int)text[i] - 1;  word = (char)wordInt;  wordDecript += word.ToString();  }  return wordDecript;  }  }  class Program  {  static void Main(string[] args)  {  string newText = Encrypter.Encrypt("CMPICA");  Console.WriteLine($"Encrypted text is: {newText}");  string TextDescripted = Encrypter.Decrypt(newText);  Console.WriteLine($"Decrypted text is: {TextDescripted}");  Console.Read();  }  }  }  Output-: |
| 12 | | A complex number has two parts: the real part and the imaginary part. In a number such as a+bi (2-3i, for example) the real part would be "a" (2) and the imaginary part would be "b" (-3).  Create a class ComplexNumber with:  A constructor to set the values for the real part and the imaginary part.  Setters and getters for both.  A method "ToString", which would return "(2,-3)"  A method "GetMagnitude" to return the magnitude of the complex number (square root of a2+b2)  A method "Add", to sum two complex numbers (the real part will be the sum of both real parts, and the imaginary part will be the sum of both imaginary parts)  Create a test program, to try these capabilities.  Code-:  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_12  {  class ComplexNumber  {  protected double realPart, imaginaryPart;  public ComplexNumber(double realPart, double imaginaryPart)  {  this.realPart = realPart;  this.imaginaryPart = imaginaryPart;  }  public double GetReal()  {  return realPart;  }  public void SetReal(double realPart)  {  this.realPart = realPart;  }  public double GetImaginary()  {  return imaginaryPart;  }  public void SetImaginary(double imaginaryPart)  {  this.imaginaryPart = imaginaryPart;  }  public new string ToString()  {  return "(" + realPart + "," + imaginaryPart + ")";  }  public double GetMagnitude()  {  return Math.Sqrt((realPart \* realPart) + (imaginaryPart \* imaginaryPart));  }  public void Add(ComplexNumber c2)  {  realPart += c2.GetReal();  imaginaryPart += c2.GetImaginary();  }  }  class Program  {  static void Main(string[] args)  {  ComplexNumber number = new ComplexNumber(10, 4);  Console.WriteLine("Number is: " + number.ToString());  number.SetImaginary(-3);  Console.WriteLine("Number is: " + number.ToString());  Console.Write("Magnitude is: ");  Console.WriteLine(number.GetMagnitude());  ComplexNumber num = new ComplexNumber(-1, 1);  number.Add(num);  Console.Write("After adding: ");  Console.WriteLine(number.ToString());  Console.Read();  }  }  }  Output-: |
| 13 | | classes for a catalog utility based on the class diagram:  It will be able to store information about music files, films and computer programs.  For each item, it must store name, code, category and size.  For films, it must also hold the director, the main actor and the main actress. For music files, the singer and the length (in seconds).  For music and movies, it must have a method "Play" (no need to implement method just declare it) and a method "RetrieveInformation", which will (no need to implement just declare it) connect to an internet server to get information about it.  D:\One Drive\OneDrive - Charotar University\CA 315 ADNF\Class Diagram For Assignment 3.png |
|  | | Code-:  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Q\_13  {  class Item  {  protected string name, code, category, size;  public Item()  {  name = "";  code = "";  category = "";  size = "";  }  public Item(string name,string code,string category,string size)  {  this.name = name;  this.code = code;  this.category = category;  this.size = size;  }  }  class Film : Item  {  protected string director, mainActor, mainActress;  public Film()  {  director = "";  mainActor = "";  mainActress = "";  }  public Film(string director,string mainActor,string mainActress)  {  this.director = director;  this.mainActor = mainActor;  this.mainActress = mainActress;  }  public void Play()  {  Console.WriteLine("The song is playing.");  }  public void Retriveinformation()  {  Console.WriteLine($"Song is Directed by {director} and MainActor is {mainActor} and MainActress is {mainActress}");  }  }  class Music : Item  {  protected string singer;  protected int length;  public Music()  {  singer = "";  length = 0;  }  public Music(string singer, int length)  {  this.singer = singer;  this.length = length;  }  public void Play()  {  Console.WriteLine("Play Function is Executes");  }  public void Retriveinformation()  {  Console.WriteLine($"Song singer is {singer} and Lengin {length} min");  }  }  class ComputerProgram : Item  {  }  class Program  {  static void Main(string[] args)  {  String s1 = "Mohan Das", s2 = "SalmanKhan", s3 = "Dipika";  Item i = new Item();  Film f = new Film(s1, s2, s3);  f.Play();  f.Retriveinformation();  Music m = new Music("SonuNigam", 5);  m.Play();  m.Retriveinformation();  Console.Read();  }  }  }  Output-: |

Thank you