**Mst. Moushin Anjum**

**ID:221O3305**

1. **Write a C# program where input type of the shape output is the area of that shape.**

**Answer to the question no:01**

using System;

class Moushin

{

static void Main()

{

Console.Write("Enter shape ");

string shape = Console.ReadLine().ToLower();

double area = 0;

if (shape == "circle")

{

Console.Write("Enter radius: ");

double r = Convert.ToDouble(Console.ReadLine());

area = Math.PI \* r \* r;

}

else if (shape == "rectangle")

{

Console.Write("Enter width: ");

double w = Convert.ToDouble(Console.ReadLine());

Console.Write("Enter height: ");

double h = Convert.ToDouble(Console.ReadLine());

area = w \* h;

}

else if (shape == "triangle")

{

Console.Write("Enter base: ");

double b = Convert.ToDouble(Console.ReadLine());

Console.Write("Enter height: ");

double h = Convert.ToDouble(Console.ReadLine());

area = 0.5 \* b \* h;

}

else

{

Console.WriteLine("Unknown shape.");

return;

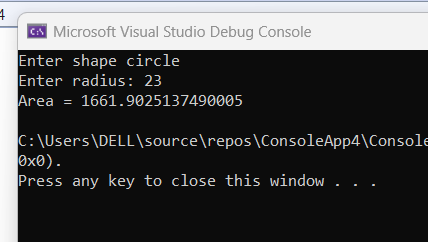
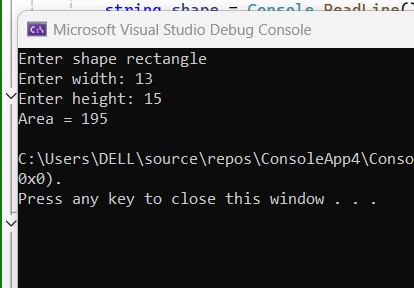
}

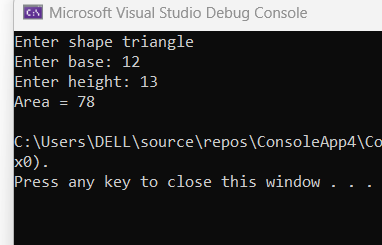
Console.WriteLine("Area = " + area);

}

}

**Output:**

****



**2. Write a C# program to create a vehicle class hierarchy. The base class should be**

**Vehicle, with subclasses Truck, Car and Motorcycle. Each subclass should have**

**properties such as model, year, and fuel type. Implement methods for calculating**

**fuel efficiency, distance travelled, and maximum speed.**

**Answer to the question no:02**

using System;

class Vehicle

{

public string Model { get; set; }

public int Year { get; set; }

public string FuelType { get; set; }

public Vehicle(string model, int year, string fuelType)

{

Model = model;

Year = year;

FuelType = fuelType;

}

public virtual double CalculateFuelEfficiency()

{

return 0;

}

public virtual double CalculateDistanceTravelled(double hours, double speed)

{

return speed \* hours;

}

public virtual double GetMaxSpeed()

{

return 0;

}

public void DisplayInfo()

{

Console.WriteLine($"Model: {Model}, Year: {Year}, Fuel Type: {FuelType}");

Console.WriteLine($"Fuel Efficiency: {CalculateFuelEfficiency()} km/l");

Console.WriteLine($"Max Speed: {GetMaxSpeed()} km/h");

}

}

class Truck : Vehicle

{

public Truck(string model, int year, string fuelType)

: base(model, year, fuelType)

{

}

public override double CalculateFuelEfficiency()

{

return 5;

}

public override double GetMaxSpeed()

{

return 120;

}

}

class Car : Vehicle

{

public Car(string model, int year, string fuelType)

: base(model, year, fuelType)

{

}

public override double CalculateFuelEfficiency()

{

return 15;

}

public override double GetMaxSpeed()

{

return 180;

}

}

class Motorcycle : Vehicle

{

public Motorcycle(string model, int year, string fuelType)

: base(model, year, fuelType)

{

}

public override double CalculateFuelEfficiency()

{

return 25;

}

public override double GetMaxSpeed()

{

return 160;

}

}

class Program

{

static void Main()

{

Vehicle truck = new Truck("F-150", 2024, "Diesel");

Vehicle car = new Car("Toyota", 2018, "Petrol");

Vehicle motorcycle = new Motorcycle("Yamaha R1", 2025, "Petrol");

Console.WriteLine("Truck Info:");

truck.DisplayInfo();

Console.WriteLine($"Distance travelled in 3 hours at 50 km/h: {truck.CalculateDistanceTravelled(3, 50)} km");

Console.WriteLine();

Console.WriteLine("Car Info:");

car.DisplayInfo();

Console.WriteLine($"Distance travelled in 2 hours at 100 km/h: {car.CalculateDistanceTravelled(2, 100)} km");

Console.WriteLine();

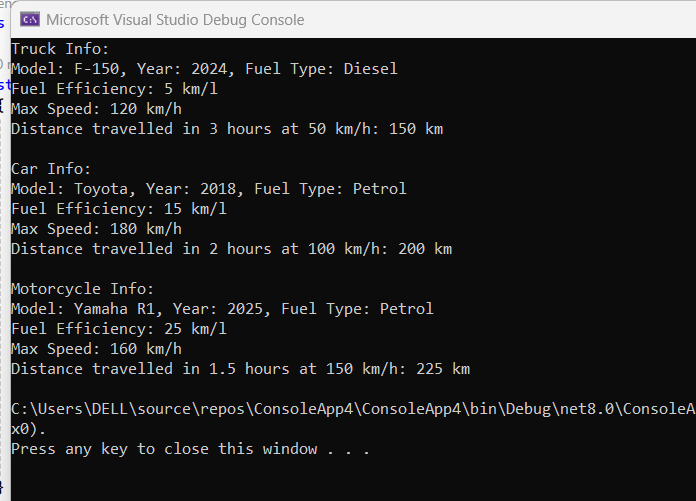
Console.WriteLine("Motorcycle Info:");

motorcycle.DisplayInfo();

Console.WriteLine($"Distance travelled in 1.5 hours at 150 km/h: {motorcycle.CalculateDistanceTravelled(1.5, 150)} km");

}

}



**Output:**

**3. Write a C# program where the bases class is shape. And the derived classes are of**

**different shape. (Circle, Rectangle, Triangle etc.) The member method of the the**

**derived classes are going to calculate the area of the shape.**

**For example,**

**Circle class will calculate the area of the circle, rectangle class or triangle will also**

**do the same.**

**Answer to the question no:03**

using System;

class Shape

{

public virtual double GetArea()

{

return 0;

}

}

class Circle : Shape

{

public double Radius;

public Circle(double r)

{

Radius = r;

}

public override double GetArea()

{

return 3.14 \* Radius \* Radius;

}

}

class Rectangle : Shape

{

public double Width, Height;

public Rectangle(double w, double h)

{

Width = w;

Height = h;

}

public override double GetArea()

{

return Width \* Height;

}

}

class Triangle : Shape

{

public double BaseLength, Height;

public Triangle(double b, double h)

{

BaseLength = b;

Height = h;

}

public override double GetArea()

{

return 0.5 \* BaseLength \* Height;

}

}

class Program

{

static void Main()

{

Circle circle = new Circle(6);

Rectangle rectangle = new Rectangle(5, 7);

Triangle triangle = new Triangle(4, 8);

Console.WriteLine("Circle Area: " + circle.GetArea());

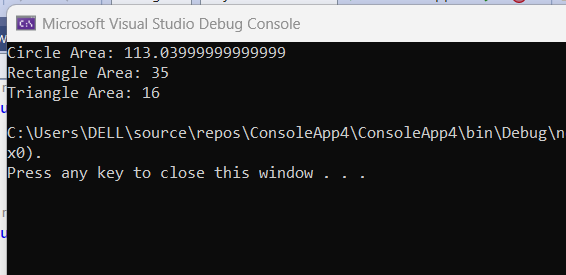
Console.WriteLine("Rectangle Area: " + rectangle.GetArea());

Console.WriteLine("Triangle Area: " + triangle.GetArea());

}

}

**Output:**

****

**4. Write a C# Program with a class called Inherit which should have a String type**

**variable named str and a member method with the name inheritmethod.**

**a) Create a child class of Inherit with the name InheritChild and access**

**the str from Inherit class. and it will have its own method called inheritchildmethod**

**b) Create a child class of InheritChild with the name Child and from this class**

**access all the property from it's parents.**

**Answer to the question no:04**

using System;

class Inherit

{

public string str = "Inherit class";

public void inheritmethod()

{

Console.WriteLine(str);

}

}

class InheritChild : Inherit

{

public void inheritchildmethod()

{

Console.WriteLine("This is InheritChild class.");

Console.WriteLine("Accessing str: " + str);

}

}

class Child : InheritChild

{

public void childmethod()

{

Console.WriteLine("This is Child class.");

Console.WriteLine("Accessing str: " + str);

inheritmethod();

inheritchildmethod();

}

}

class Program

{

static void Main()

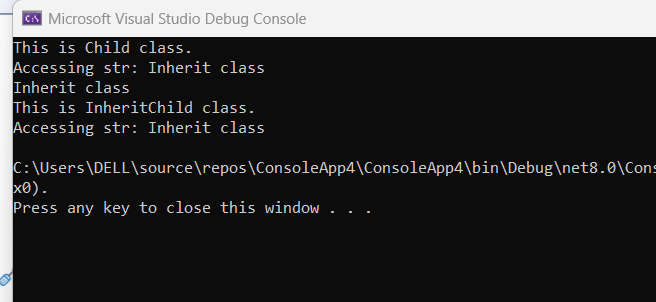
{

Child child = new Child();

child.childmethod();

}

}

**Output:**

**5. Write a C# program with a class called Animal which will have some member**

**method as walk and eat.**

**a) Create a child class of Animal called Dog with the property nooflegs and bark**

**b) Create a child class of Animal called Bird with the property noofwings and fly**

**c) Create a child class of Bird called Dove with the property color**

**Answer to the question no:05**

using System;

class Animal

{

public void walk()

{

Console.WriteLine("Animal can walk");

}

public void eat()

{

Console.WriteLine("Animal can eat");

}

}

class Dog : Animal

{

public int noOfLegs = 4;

public void bark()

{

Console.WriteLine("Dog can bark");

}

}

class Bird : Animal

{

public int noOfWings = 2;

public void fly()

{

Console.WriteLine("Bird can fly");

}

}

class Dove : Bird

{

public string color = "White";

public void showColor()

{

Console.WriteLine("Dove color is: " + color);

}

}

class Program

{

static void Main()

{

Dog dog = new Dog();

dog.walk();

dog.eat();

dog.bark();

Console.WriteLine("Dog has " + dog.noOfLegs + " legs\n");

Dove dove = new Dove();

dove.walk();

dove.eat();

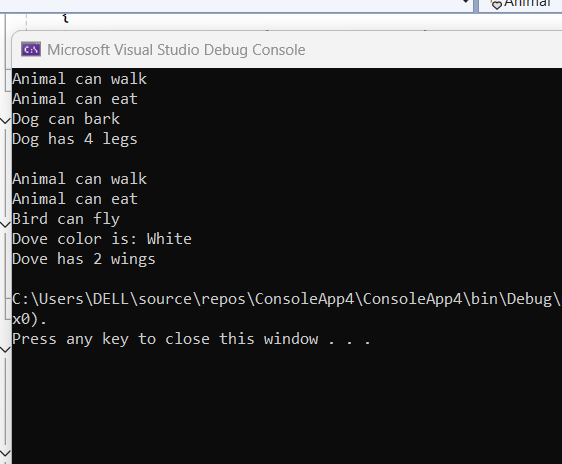
dove.fly();

dove.showColor();

Console.WriteLine("Dove has " + dove.noOfWings + " wings");

}

}

**Output:**