**Task-3**

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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Problem\_1

{

abstract class Shape

{

public abstract double CalculateArea();

}

class Circle : Shape

{

private double radius;

public Circle(double radius)

{

this.radius = radius;

}

public override double CalculateArea()

{

return Math.PI \* radius \* radius;

}

}

class Rectangle : Shape

{

private double width, height;

public Rectangle(double width, double height)

{

this.width = width;

this.height = height;

}

public override double CalculateArea()

{

return width \* height;

}

}

class Triangle : Shape

{

private double baseLength, height;

public Triangle(double baseLength, double height)

{

this.baseLength = baseLength;

this.height = height;

}

public override double CalculateArea()

{

return 0.5 \* baseLength \* height;

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the shape type (circle / rectangle / triangle):");

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

Shape shape = null;

switch (shapeType)

{

case "circle":

Console.Write("Enter radius: ");

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

shape = new Circle(radius);

break;

case "rectangle":

Console.Write("Enter width: ");

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

Console.Write("Enter height: ");

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

shape = new Rectangle(width, height);

break;

case "triangle":

Console.Write("Enter base: ");

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

Console.Write("Enter height: ");

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

shape = new Triangle(baseLength, triHeight);

break;

default:

Console.WriteLine("Unknown shape type.");

return;

}

double area = shape.CalculateArea();

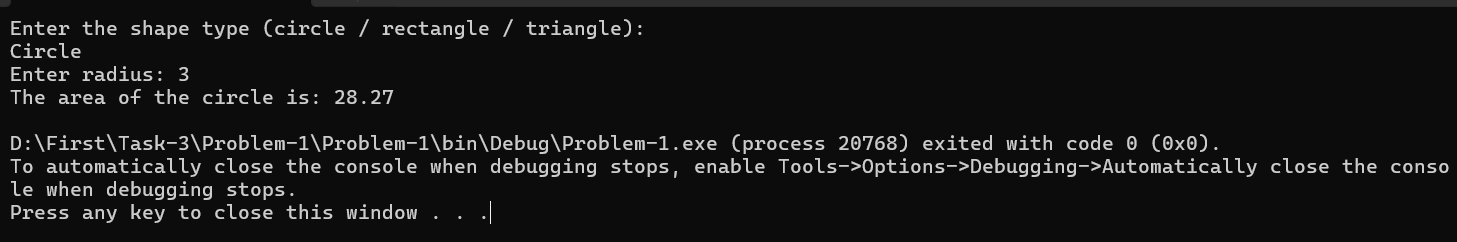
Console.WriteLine($"The area of the {shapeType} is: {area:F2}");

}

}

}

Output:



**Problem-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.

**Ans:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Problem\_2

{

abstract 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 abstract double CalculateFuelEfficiency();

public abstract double CalculateDistanceTravelled(double fuel);

public abstract double GetMaxSpeed();

public void DisplayInfo()

{

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

}

}

class Car : Vehicle

{

public Car(string model, int year, string fuelType) : base(model, year, fuelType) { }

public override double CalculateFuelEfficiency()

{

return 15.0;

}

public override double CalculateDistanceTravelled(double fuel)

{

return fuel \* CalculateFuelEfficiency();

}

public override double GetMaxSpeed()

{

return 180;

}

}

class Truck : Vehicle

{

public Truck(string model, int year, string fuelType) : base(model, year, fuelType) { }

public override double CalculateFuelEfficiency()

{

return 8.0;

}

public override double CalculateDistanceTravelled(double fuel)

{

return fuel \* CalculateFuelEfficiency();

}

public override double GetMaxSpeed()

{

return 120;

}

}

class Motorcycle : Vehicle

{

public Motorcycle(string model, int year, string fuelType) : base(model, year, fuelType) { }

public override double CalculateFuelEfficiency()

{

return 35.0;

}

public override double CalculateDistanceTravelled(double fuel)

{

return fuel \* CalculateFuelEfficiency();

}

public override double GetMaxSpeed()

{

return 160;

}

}

class Program

{

static void Main(string[] args)

{

Vehicle car = new Car("Toyota Corolla", 2022, "Petrol");

Vehicle truck = new Truck("Volvo Truck", 2020, "Diesel");

Vehicle bike = new Motorcycle("Yamaha R15", 2023, "Petrol");

Vehicle[] vehicles = { car, truck, bike };

foreach (var vehicle in vehicles)

{

Console.WriteLine("\nVehicle Information:");

vehicle.DisplayInfo();

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

Console.WriteLine($"Distance Travelled with 10L: {vehicle.CalculateDistanceTravelled(10)} km");

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

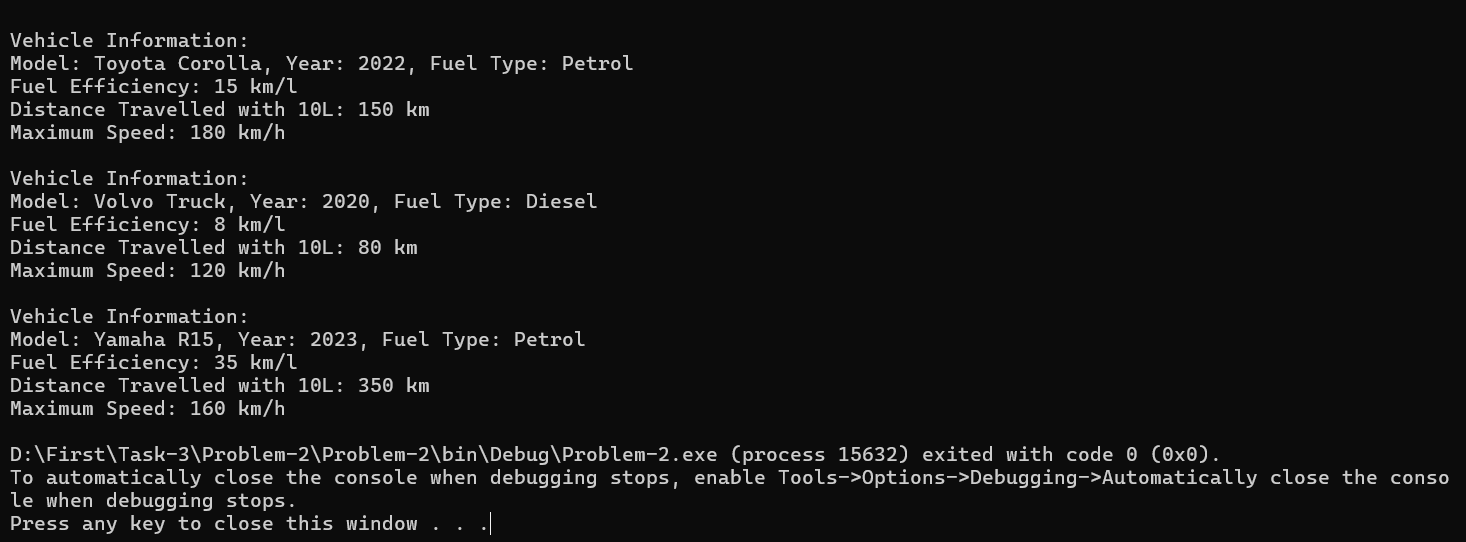
}

}

}

}

**Output:**



**Problem-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.

Ans:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Problem\_3

{

abstract class Shape

{

public abstract double CalculateArea();

}

class Circle : Shape

{

private double radius;

public Circle(double radius)

{

this.radius = radius;

}

public override double CalculateArea()

{

return Math.PI \* radius \* radius;

}

}

class Rectangle : Shape

{

private double width, height;

public Rectangle(double width, double height)

{

this.width = width;

this.height = height;

}

public override double CalculateArea()

{

return width \* height;

}

}

class Triangle : Shape

{

private double baseLength, height;

public Triangle(double baseLength, double height)

{

this.baseLength = baseLength;

this.height = height;

}

public override double CalculateArea()

{

return 0.5 \* baseLength \* height;

}

}

class Program

{

static void Main(string[] args)

{

Console.Write("Enter radius of the circle: ");

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

Shape circle = new Circle(radius);

Console.WriteLine($"Area of Circle: {circle.CalculateArea():F2}");

Console.Write("\nEnter width of the rectangle: ");

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

Console.Write("Enter height of the rectangle: ");

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

Shape rectangle = new Rectangle(width, height);

Console.WriteLine($"Area of Rectangle: {rectangle.CalculateArea():F2}");

Console.Write("\nEnter base of the triangle: ");

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

Console.Write("Enter height of the triangle: ");

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

Shape triangle = new Triangle(baseLength, triHeight);

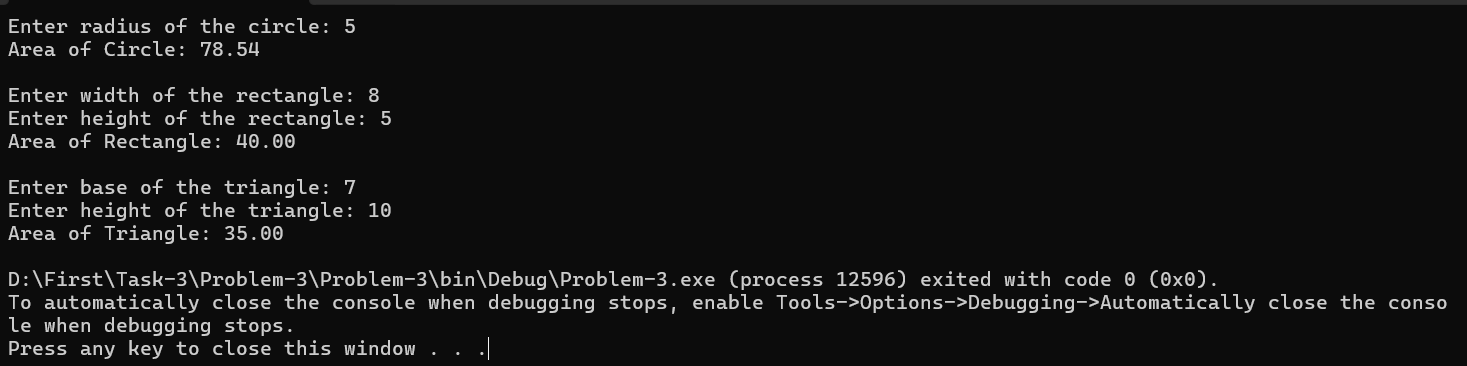
Console.WriteLine($"Area of Triangle: {triangle.CalculateArea():F2}");

}

}

}

Output:



**Problem-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.

**Ans:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Problem\_4

{

class Inherit

{

protected string str = "Hello from Inherit class";

public void inheritmethod()

{

Console.WriteLine("Inherit Method: " + str);

}

}

class InheritChild : Inherit

{

public void inheritchildmethod()

{

Console.WriteLine("InheritChild Method: Accessing str = " + str);

}

}

class Child : InheritChild

{

public void childMethod()

{

Console.WriteLine("Child Method: Accessing str = " + str);

}

}

class Program

{

static void Main(string[] args)

{

Child childObj = new Child();

childObj.inheritmethod();

childObj.inheritchildmethod();

childObj.childMethod();

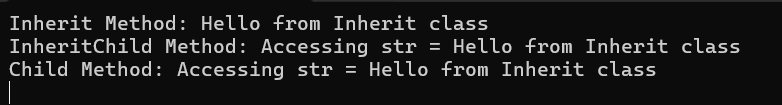
Console.ReadLine();

}

}

}

**Output:**



**Problem-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

Ans:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Problem\_5

{

class Animal

{

public void Walk()

{

Console.WriteLine("Animal is walking.");

}

public void Eat()

{

Console.WriteLine("Animal is eating.");

}

}

class Dog : Animal

{

public int NoOfLegs { get; set; }

public void Bark()

{

Console.WriteLine("Dog is barking.");

}

}

class Bird : Animal

{

public int NoOfWings { get; set; }

public void Fly()

{

Console.WriteLine("Bird is flying.");

}

}

class Dove : Bird

{

public string Color { get; set; }

public void ShowInfo()

{

Console.WriteLine($"Dove color is {Color} and it has {NoOfWings} wings.");

}

}

class Program

{

static void Main(string[] args)

{

Dog dog = new Dog();

dog.NoOfLegs = 4;

Console.WriteLine($"Dog has {dog.NoOfLegs} legs.");

dog.Walk();

dog.Eat();

dog.Bark();

Console.WriteLine();

Dove dove = new Dove();

dove.NoOfWings = 2;

dove.Color = "White";

dove.Walk();

dove.Eat();

dove.Fly();

dove.ShowInfo();

Console.ReadLine();

}

}

}

**Output:**

