Name: Aryan Patel

SIN : 301226774

Driver class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Week\_06\_lab\_16

{

class Program

{

static void Main(string[] args)

{

//although Shape is an abstract is can be used as a reference type

//any child class of Shape is also a Shape

List<Shape> shapes = new List<Shape>();

shapes.Add(new Square("s1", 2));

shapes.Add(new Rectangle("r1", 2, 3));

shapes.Add(new Circle("c1", 2));

shapes.Add(new Triangle("t1", 4, 6));

shapes.Add(new Ellipse("e1", 2, 3));

shapes.Add(new Diamond("d1", 2, 3));

shapes.Add(new Square("s2", 5));

shapes.Add(new Rectangle("r2", 5, 4));

shapes.Add(new Circle("c2", 1));

shapes.Add(new Triangle("t2", 7, 8));

foreach (var s in shapes)

{

Console.WriteLine(s);

}

}

}

}

Shape class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Week\_06\_lab\_16

{

abstract class Shape

{

public string Name { get; protected set; }

public abstract double Area { get; }

public Shape(string name)

{

this.Name = name;

}

public override string ToString()

{

return $"Shape: {this.Name} (Area: {this.Area})\n";

}

}

}

Square class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Week\_06\_lab\_16

{

class Square : Shape

{

public double Length { get; set; }

private double \_area;

public override double Area {

get

{

return this.\_area;

}

}

public Square(string name, double length) : base (name)

{

this.Length = length;

this.\_area = Math.Pow(length, 2);

}

}

}

Rectangle class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Week\_06\_lab\_16

{

class Rectangle : Shape

{

private double \_area;

public override double Area { get => this.\_area; }

public double Width { get; set; }

public double Height { get; set; }

public Rectangle(string name, double height, double width): base(name)

{

this.Width = width;

this.Height = height;

this.\_area = this.Width \* this.Height;

}

}

}

Circle class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Week\_06\_lab\_16

{

class Circle : Square

{

private double \_area;

public override double Area

{

get

{

return this.\_area;

}

}

public Circle(string name, double length) : base(name,length)

{

this.\_area = Math.PI \* length \* length;

}

}

}

Diamond class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Week\_06\_lab\_16

{

class Diamond : Rectangle

{

private double \_area;

public override double Area => this.\_area;

public Diamond(string name, double height, double width) : base(name, height, width)

{

this.\_area = width \* height \* 0.5;

}

}

}

Triangle class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Week\_06\_lab\_16

{

class Triangle : Rectangle

{

private double \_area;

public override double Area => this.\_area;

public Triangle(string name, double height, double width) : base(name,height,width)

{

this.\_area = width \* height \* 0.5;

}

}

}

Ellipse class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Week\_06\_lab\_16

{

class Ellipse: Rectangle

{

private double \_area;

public override double Area => this.\_area;

public Ellipse(string name, double height, double width): base(name,height,width)

{

this.\_area = Math.PI \* width \* height;

}

}

}