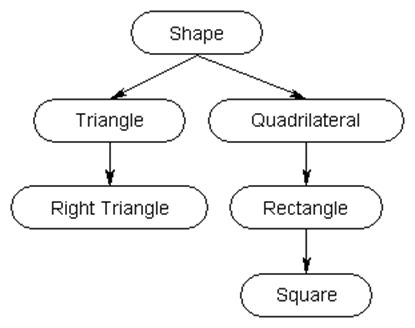
**Assignment:**

**Implement the concept of OOPs in following diagram.**

****

**Codes:**

**class Shape**

{

int length;

int breadth;

int height;

float result;

public void perimeter()

{

Console.WriteLine("\n\nRectangle");

Console.WriteLine("Enter Length:");

length = int.Parse(Console.ReadLine());

Console.WriteLine("Enter breadth:");

breadth = int.Parse(Console.ReadLine());

result = 2\*(length + breadth);

}

public void perimeter(char square,char Square)

{

Console.WriteLine("\n\nSquare");

Console.WriteLine("Enter Length:");

length = int.Parse(Console.ReadLine());

result = 4 \* length;

}

public void perimeter(char triangle)

{

Console.WriteLine("\n\nTriangle");

Console.WriteLine("Enter Length:");

length = int.Parse(Console.ReadLine());

Console.WriteLine("Enter breadth:");

breadth = int.Parse(Console.ReadLine());

Console.WriteLine("Enter height:");

height = int.Parse(Console.ReadLine());

result = length+breadth+height;

}

public void perimeter(int triangle)

{

Console.WriteLine("\n\nRight angle Triangle");

Console.WriteLine("Enter Length:");

length = int.Parse(Console.ReadLine());

Console.WriteLine("Enter breadth:");

breadth = int.Parse(Console.ReadLine());

Console.WriteLine("Enter height:");

height = int.Parse(Console.ReadLine());

result = length + breadth + height;

}

public float Display()

{

return result;

}

}

}

**class Triangle : Shape**

{

public Triangle(int rt)

{

base.perimeter(1);

}

public Triangle()

{

base.perimeter('t');

}

public float displayofRT()

{

return base.Display();

}

public new void Display()

{

Console.WriteLine("The Perimeter is {0}",base.Display());

}

}

}

**class Right\_Triangle : Triangle**

{

public Right\_Triangle() : base(0) { }

public new void Display()

{

Console.WriteLine("The Perimeter is {0}", base.displayofRT());

}

}

}

**class Quadrilateral:Shape**

{

public Quadrilateral(char s)

{

base.perimeter('s','s');

}

public Quadrilateral()

{

base.perimeter();

}

public new float Display()

{

return base.Display();

}

}

}

**class Rectangle:Quadrilateral**

{

public Rectangle(char s)

:base('s'){}

public Rectangle()

: base(){}

public float DisplayOfSquare()

{

return base.Display();

}

public new void Display()

{

Console.WriteLine("The perimeter is {0}",base.Display());

}

}

}

**class Square:Rectangle**

{

public Square():base('s'){}

public new void Display()

{

Console.WriteLine("The perimeter is {0}", base.DisplayOfSquare());

}

}

}

**class Program**

{

static void Main(string[] args)

{

Square square = new Square();

square.Display();

Rectangle rectangle = new Rectangle();

rectangle.Display();

Right\_Triangle Rt = new Right\_Triangle();

Rt.Display();

Triangle triangle = new Triangle();

triangle.Display();

}

}

}

**Output:**

