

Nullable variable

```
using System;
using System.Collections.Generic;
using System.Text;

public class Vector {
    public double? R = null;
    public double? Theta = null;

    public double? ThetaRadians {
        get {
            return (Theta * Math.PI / 180.0);
        }
    }

    public Vector(double? r, double? theta) {
        if (r < 0) { r = -r; theta += 180; } theta = theta % 360; R = r; Theta = theta; }
        public static Vector operator +(Vector op1, Vector op2) { try { double newX = op1.R.Value * Math.Sin(op1.ThetaRadians.Value) + op2.R.Value * Math.Sin(op2.ThetaRadians.Value); double newY = op1.R.Value * Math.Cos(op1.ThetaRadians.Value) + op2.R.Value * Math.Cos(op2.ThetaRadians.Value); double newR = Math.Sqrt(newX * newX + newY * newY); double newTheta = Math.Atan2(newX, newY) * 180.0 / Math.PI; return new Vector(newR, newTheta); }
        catch { return new Vector(null, null); } }
        public static Vector operator -(Vector op1) { return new Vector(-op1.R, op1.Theta); }
        public static Vector operator -(Vector op1, Vector op2) { return op1 + (-op2); }
        public override string ToString() { string rString = R.HasValue ? R.ToString() : "null"; string thetaString = Theta.HasValue ? Theta.ToString() : "null"; return string.Format("({0}, {1})", rString, thetaString); }
    }

    class Program {
        public static void Main(string[] args) { Vector v1 = GetVector("vector1"); Vector v2 = GetVector("vector2"); Console.WriteLine("{0} + {1} = {2}"); }
        static Vector GetVector(string name) { Vector vector = new Vector(); Console.Write("Enter the value of R for " + name + ": "); if (double.TryParse(Console.ReadLine(), out double r)) { vector.R = r; } else { Console.WriteLine("Invalid input for R"); }
        Console.Write("Enter the value of Theta for " + name + ": "); if (double.TryParse(Console.ReadLine(), out double theta)) { vector.Theta = theta; } else { Console.WriteLine("Invalid input for Theta"); }
        return vector; }
    }
}
```

```
{2}", v1, v2, v1 + v2); Console.WriteLine("{0} - {1} = {2}",  
v1, v2, v1 - v2); Console.ReadKey(); } public static Vector  
GetVector(string name) { Console.WriteLine("Input {0}  
magnitude:", name); double? r = GetNullableDouble();  
Console.WriteLine("Input {0} angle (in degrees):", name);  
double? theta = GetNullableDouble(); return new Vector(r,  
theta); } public static double? GetNullableDouble() { double?  
result; string userInput = Console.ReadLine(); try { result =  
double.Parse(userInput); } catch { result = null; } return  
result; } } [/csharp]
```