The two principal kinds of coordinate systems used in algebra are the Cartesian (or xy) coordinate system and the Polar (or r-theta) coordinate system.

### Cartesian to Polar
- \( r = \sqrt{x_1^2 + y_1^2} \)
- \( \theta = \tan^{-1}\left(\frac{y_1}{x_1}\right) \)

### Polar to Cartesian
- \( x_1 = r \cdot \cos(\theta) \)
- \( y_1 = r \cdot \sin(\theta) \)

The point P1 is often taken to represent the end point of a vector from the origin to P1. The vector is said to have a magnitude of r and an angle of \( \theta \). Vectors can be added, subtracted, multiplied, or divided. Addition and subtraction are most easily done in the Cartesian coordinate system and multiplication and division are most easily done in the Polar coordinate system.

### Addition
If \( V = V1 + V2 \), then \( x_v = x_1 + x_2 \) and \( y_v = y_1 + y_2 \)

### Subtraction
If \( V = V1 - V2 \), then \( x_v = x_1 - x_2 \) and \( y_v = y_1 - y_2 \)

### Multiplication
If \( V = V1*V2 \), then \( r_v = r_1 * r_2 \) and \( \theta_v = \theta_1 + \theta_2 \)

### Division
If \( V = V1/V2 \), then \( r_v = r_1/r_2 \) and \( \theta_v = \theta_1 - \theta_2 \)

Add four public member functions to the Polar class on the following page for **Addition**, **Subtraction**, **Multiplication**, and **Division**. Your code should correctly execute the following main code statements:
main()
{
Polar a(1, 45), b(0.5, 30);
Polar c;
c = a.Add(b);
c.Prt();
c = a.Subtract(b);
c.Prt();
c = a.Multiply(b);
c.Prt();
c = a.Divide(b);
c.Prt();
return 0;
}

The correct answers are
Addition: r = 1.98, θ = 37.55°
Subtraction: r = .261, θ = 127.53°
Multiplication: r = 0.5, θ = 75°
Division: r = 2.0, θ = 15°
//polar .cpp
// This is the main program that exercises the polar class.
#include<iostream>
#include<stdlib.h>
using namespace std;

class Polar
{public:
   Polar(double = 0, double = 0); //Constructor
   ~Polar(); //Destructor
   void set(double, double);
   void prt();
private:
   double r, theta;
};

int main()
{Polar a(1, 45), b(0.5, 30);
Polar c;
c = a.Add(b);
c.Prt();
c = a.Subtract(b);
c.Prt();
c = a.Multiply(b);
c.Prt();
c = a.Divide(b);
c Пгт();
system("PAUSE");
return 0;
}

// constructor
Polar::Polar(double a, double b)
{set(a, b);
}

// destructor
Polar::~Polar()
{cout << "Polar class destroyed." << endl;
}

// set
void Polar::set(double a, double b)
{r = a;
   theta = b;
}

// prt
void Polar::prt()
{cout << "r = " << r << " theta = " << theta << endl;}