

```

/*
File: polarcoord.cpp
Created by: Tan Qi Hao
Creation Date: 3/15/2019
Synopsis: This program reads the polar coordinates of a point and outputs
the cartesian coordinates.
*/

#include <iostream>
#include <cmath>

using namespace std;

// FUNCTION PROTOTYPE FOR degrees2radians
// Parameter D is the degree of an angle
double degrees2radians(double D);

// FUNCTION PROTOTYPE FOR compute_coord
/* There are 4 parameters in this function prototype.
Parameter radius is the polar radius of the point.
Parameter angle_radians is the polar angle of the point.
Parameter x is the x-coordinate of the point.
Parameter y is the y-coordinate of the point.
*/
void compute_coord(double radius, double angle_radians, double & x, double
& y);

// DO NOT MODIFY THE MAIN ROUTINE IN ANY WAY
int main()
{
    double angle_degrees(0.0), angle_radians(0.0), radius(0.0);
    double coord_x(0.0), coord_y(0.0);

    // Read in polar coordinates
    cout << "Enter radius: ";
    cin >> radius;

    cout << "Enter polar angle (degrees): ";
    cin >> angle_degrees;

    // Convert degrees to radians
    angle_radians = degrees2radians(angle_degrees);

    // Compute Cartesian (x,y) coordinates
    compute_coord(radius, angle_radians, coord_x, coord_y);

    // Output Cartesian coordinates
    cout << "Cartesian coordinates: ";
    cout << "(" << coord_x << ", " << coord_y << ")" << endl;

    return 0;
}

// DEFINE FUNCTION degrees2radians here:

```

```

// This function converts degrees into radians.
double degrees2radians(double D){

    double R; //R is the radians

    R = D * (M_PI / 180);

    return R;

}

// DEFINE FUNCTION compute_coord here:
/* This function converts polar coordinates into Cartesian coordinates
   of a point.
*/
void compute_coord(double radius, double angle_radians, double & x, double
& y){

    x = radius * cos(angle_radians);
    y = radius * sin(angle_radians);
}

```