





//Shrabanti Basu

//Feb 22, 2016

//Program 3

//This program illustrates the use of files for input and output, use cmath library

//and format numerical output.

#include <iostream>

#include <fstream> //for file stream objects

#include <cmath> //for math functions

#include <string> //for string class

#include <iomanip> //for output formatting

using namespace std;

int main()

{

cout << "Shrabanti Basu\n";

cout << "February 22, 2016\n";

cout << "Program 3\n";

cout << "This program illustrates the use of files for input and output,\n";

cout << "use of cmath library and format numerical output.\n\n";

cout << "The program asks user to enter the filename,\n";

cout << "reads file information, x and y values of Cartesian co-ordinates\n";

cout << "converts them into Polar co-ordinate format.\n";

cout << "The result is printed to the console and written to an output file.\n";

cout << "The user enters input file name, and name of the output file is specified in the program.\n\n";

const float PI = 3.1415926; //declare constant variable PI

//create ifstream and ofstream objects

ifstream inputFile;

ofstream outputFile;

string fileName, //to store filename entered by user

inputLine; //to store file information read from input file

float x, y, //to read x and y co-ordinates from the input file

radius, angleRadian, //to store radius 'r' and angle in radians for polar co-ordinates

angleDegree; //to store the degree equivalent of the angle in radians

cout << "Enter the input file name: "; //get filename from user to the open the file

cin >> fileName;

inputFile.open(fileName);

//check if input file has opened properly before reading input

if (inputFile)

{

getline(inputFile, inputLine); //read file information from the first line

inputFile >> x; //read x coord

inputFile >> y; //read y coord

radius = sqrt(pow(x, 2.0) + pow(y, 2.0)); //calculate radius

angleRadian = atan2(y, x); //calculate the angle in radians

angleDegree = angleRadian \* (180 / PI); //calculate the angle in degrees

cout << setprecision(4) << fixed << showpoint; //print 4 decimal places

//print inputfile name and information to console

cout << "\n\nInput File: " << fileName << endl;

cout << inputLine << endl;

cout << "x co-ordinate: " << x << endl; //print x coord

cout << "y co-ordinate: " << y << endl; //print y coord

cout << "\nThe corresponding values in polar co-ordinates are: \n";

cout << "raduis r : " << radius << endl;

cout << "Angle in Radians: " << angleRadian << endl;

cout << "Angle in Degrees: " << angleDegree << endl << endl;

//open output file and print the similar information to it

outputFile.open("Program3OutputFile.txt"); //filename used in ofstream object as string literal

//check if outputfile had any error

if (outputFile.fail())

{

cout << "Error opening output file.\n";

}

else

{

outputFile << setprecision(4) << fixed << showpoint;

cout << "Writing data to output file Program3OutputFile.txt .....\n";

outputFile << "This file contains polar co-ordinates to corresponding Cartesian values.\n\n";

outputFile << "x co-ordinate: " << x << endl;

outputFile << "y co-ordinate: " << y << endl;

outputFile << "\nThe corresponding polar co-ordinates are: \n";

outputFile << "raduis r : " << radius << endl;

outputFile << "Angle in Radians: " << angleRadian << endl;

outputFile << "Angle in Degrees: " << angleDegree << endl << endl;

cout << "Writing to the file complete!\n\n";

outputFile.close(); //close output file

}

inputFile.close(); //close input file

}

else

{

cout << "Error opening file.\n";

}

return 0;

}