

Homework 1: Point2D

cs244

Due: September 18, 8:00 AM

Allowances, if any, for late work and related penalties will be as described in the syllabus.

Overview

Explore and acquaint yourself with Ubuntu Linux, g++, and get back into programming mode. You will implement a class to represent 2D points in space and perform some basic operations on and with them.

Topics: C++ classes, UML diagrams, Code Testing, Peer Reviews

Preparation

Step 1: Create a Folder

In Linux in your `.../Documents/Programs` folder

Create a folder named **A01**

This is necessary as you will compress the A01 folder and its contents for submission.

Step 2: Check D2L

There may be starter code for this assignment available on D2L.

Files to Create

The files for this assignment **MUST** be named:

point2D.cpp **point2D.h** **point2Dmain.cpp**
README.txt

Your *main* function is to be located in `point2Dmain.cpp`.

All code files should have comments with the file name, your name, the last modified date, and a description of the file contents. Each function should be appropriately commented in the header and implementation files – most should include pre and post conditions. The body of each function may need comments for the less than obvious parts of the code (if any).

The final submission of files should include a file named: **README.txt**

It provides instructions on how to compile/link your program (EX: `g++ file1.cpp file2.cpp`)

It also provides instructions on how to run your program (EX: after compile type: `./a.out`)

Homework 1: Point2D

cs244

Assignment Description

A point in the x - y plane is represented by its x -coordinate and y -coordinate. Design and implement a class, **point2D**, that can store and process a point in the x - y plane.

The class must allow the following operations to be performed:

- Set the coordinates of the point (x and y , both at the same time)
- Print the coordinates to the screen
as 2 values in parentheses separated by a comma, EX: (4.567, 12.345)
- Return the x coordinate
- Return the y coordinate
- Calculate the distance a point is from a second point

Your tester file must have a global function named *AskForData* that takes a `point2D` parameter variable as pass-by-reference.

It then asks the user to enter an x -coordinate, a y -coordinate and assigns them to the parameter variable.

UML Description of class point2D

point2D
- m_xCoord : double - m_yCoord : double
+ point2D(double x = 0.0, double y = 0.0) + setPoint(double x, double y) :void + print() const :void + getX() const :double + getY() const :double + calcDistTo(point2D * otherPoint) :double
Friend Functions None required

Recall

- + means public and
- means private

Homework 1: Point2D

cs244

Tester File

Design your main function, in the point2Dmain.cpp file, to demonstrate and test the capabilities of your point2D class.

```
// add code and comments as needed

void AskForData(point2D &thePoint);

void main()
{
    point2D p1, p2; //Create two point objects
    cout << "P1 -> "; AskForData(p1); //Ask to enter data from console
    cout << "P2 -> "; AskForData(p2); //Ask to enter data from console
    cout << "P1 is at: " << p1.print() << endl;
    cout << "P2 is at: " << p2.print() << endl;
    cout << "The distance between P1 and P2 is:" << p1.Distance(&p2) << endl;

    // add more test code as needed

    cout << "Program ends" << endl;
}

// Add comments for function
void AskForData(point2D &thePoint)
{
    // add code and comments as needed
}
```

Example Tester File, point2Dmain.cpp --- incomplete

makefile

N/A – this assignment does not require a makefile
instead

To Compile at the Command Line Type:

```
g++ point2D.cpp point2Dmain.cpp
```

To Run the executable created at the Command Line Type: `./a.out`

Grading

100 points possible

Assuming it is turned in on time:

Your program will be peer reviewed and evaluated by 2 members of your class.

Each will assign a grade to it from 0 to 100.

The instructor will also evaluate your work, scoring it from 0 to 100.

The final grade will then be assigned by the instructor based on the three scores.

Grading will be based on compilability, runability, correctness, comments, programming style, and similar attributes. Further details may be given in class.

Homework 1: Point2D

cs244

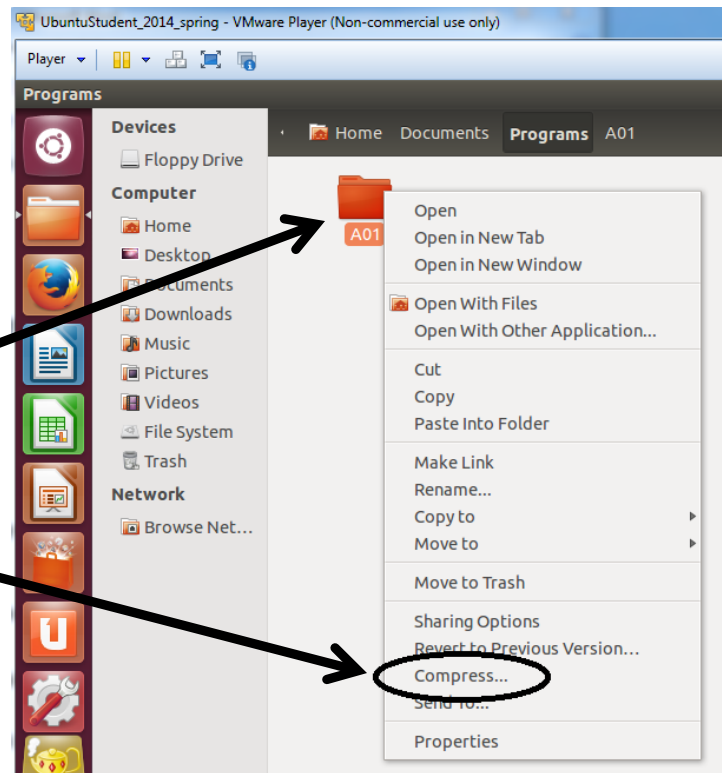
Turn-In Directions

Correctly submitting your work is worth 0 points, BUT if not done correctly will likely result in nothing to grade.

Preparation

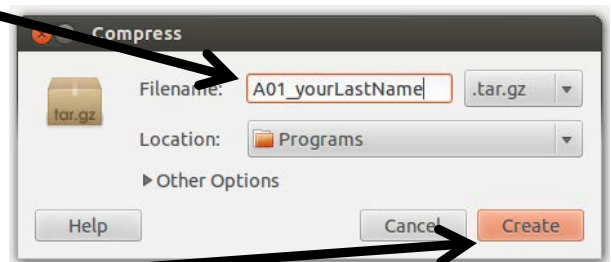
In Ubuntu Linux browse to your A01 folder
Make sure your source code files
are in the folder

**Right click on the A01 folder,
Select Compress...**



Set the file name to be A01_yourlastname.tar.gz
where *yourlastname* is your last name

Example: if your last name is Gollygee
then the filename would be
A01_Gollygee.tar.gz



Click on the Create Button

This should create the file named A01_yourlastname.tar.gz

Submit

Submit the A01_yourlastname.tar.gz file
to the correct course drop box in D2L

Peer Evaluations

For peer review evaluations you should follow the instructions given in class.
In most cases this will involve giving the above compressed files to
the peers assigned to evaluate your program.
Again in most cases, they will then review and evaluate your program.
This often means your code will need to function with a test program they created.
This emphasizes the importance of coding to specification.