

Lab #4

Classes and Polymorphism

Due: Sunday, 2/05/17, 11:59pm

(90 pts) Problem Statement: Write a C++ program that will be an information system for Oregon State University using classes as well as demonstrating a basic understanding of inheritance and polymorphism.

You will create a representation of an Oregon State University information system that will contain information about the university. The university will contain a **name** of the university, n number of **buildings**, and m number of **people**. People can be either a **student** or an **instructor**. Every person will have a **name** and an **age**. Every student will have also have a **GPA**, but an instructor will NOT have a GPA. Every instructor will have an **instructor rating**, but a student will NOT have an instructor rating. Every building will have a **name**, the **size** in sqft (preferred the real value which you need to look up), and an **address** (stored as a string, also preferred to look this up).

People will contain a method called "**do_work**" that will take in a random integer as a parameter that represents how many hours they will do work for. If the person is a student, a message will be printed to the screen that says "PERSON_NAME did X hours of homework." If the person is an instructor, a message will be printed to the screen that says "Instructor PERSON_NAME graded papers for X hours." You will need to fill in the appropriate values.

The student GPA can either be an input from the user or randomized, but it must be between 0.0 and 4.0. It cannot be preset. The instructor rating can either be an input from the user or randomized, but it must be between 0.0 and 5.0. The ages of a person can be randomized or an input, but make it realistic. You can choose whether it is randomized or user input, or both.

The university will contain a method that will **print the name and address** of all the buildings in its information system and another method that will **print the name of all the people**. The name of the university MUST be "Oregon State University" (because we are the best).

You will manually instantiate at least 1 student, 1 instructor, and 2 buildings, then give them values and store them appropriately in the university object. You can do this in whatever fashion you wish.

You will have a menu that does at least the following:

- 1) Prints names of all the buildings
- 2) Prints names of everybody at the university
- 3) Choose a person to do work
- 4) Exit the program

Note that option 3 will require you to print another menu that gives options for each person.

You may create any other functions, methods, member variables, etc. to modularize your code and complete the lab.

You may use vectors for this assignment if you so choose.

(10 pts) **Program Style/Comments**

In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Make sure you review the style guidelines for this class, and begin trying to follow them, i.e. don't align everything on the left or put everything on one line! Also view the "Things not to do in the code" page and the "Things you need to do in your code" page as you will be held to these.

Electronically submit your C++ program files (only the source files, not the executable) along with a makefile in a zip file on TEACH.

As usual, your zip file should not create a directory when unzipped and your makefile must be able to remove all files that are not .cpp or .hpp files (or any other source file) with a "make clean" command. You must also not have any memory leaks.

(10 pts) **Extra Credit**

Add an option to save the information system to a file, and add an option to read a saved information system from a file so that you can close the program, but not lose information. This will also require you to be able to add people and/or buildings to the program during runtime. This is an all or nothing extra credit (you will not get partial points for partial completion).

Rubric Breakdown

- 30% - Correct classes have been created
- 20% - Classes have correct member information
- 20% - Class methods work correctly
- 20% - Menu works correctly
- 10% - Program Style
- +10% - Extra Credit