Lab Syllabus

Artificial Intelligence

Overview

The lab projects exist to give students the chance to implement and apply concepts they have learned in lecture to solving real problems. The goal of the lab is not just to get the right answer but to get the answer by demonstrating understanding and application of the key concepts described by the course director. Students are expected to complete lab assignments on their own in order to show that they understand the idea(s) being tested.

Each lab will be in the lab repository on the studentvfiler network share and will contain a project manual, a pre-generated project file, some source code, and (in most cases) an executable example of a completed project. Students should read the project manual and finish the project by completing each of the objectives listed, using the executable example as an aide.

Due Dates

All projects are due by the beginning of the lab period following the last day of that lab project. Projects must be submitted to the LMS by the cutoff time to be given full credit.

Projects may be submitted up to two class days late for a **penalty of 25% per class day**. Incomplete projects may be submitted for partial credit. No projects will be accepted later than two class days after the date due.

The Path Planner project will span three lab days and is due at the beginning of the following lab period.

Project Submissions

As all lab projects are collected in digital format, it is **extremely important** that the guidelines are followed. Failure to follow these instructions will result in penalties.

1) Code most compile and run in debug and release mode. Debug information should never released in the final version of a software project. **Projects that do not compile will be marked zero.**

2) The public and protected interfaces of all classes must be unchanged. In addition, the interfaces of existing private methods and/or member variables may not be changed. **Students may only add private helper methods and variables.** A student may, at his option, add private inner classes.

3) Include **only those files** specified by the lab documents (relevant source and/or header files) in your archive. Projects should have **no directory structure** except as explicitly mentioned in the documentation (i.e., relevant files and folders should be submitted in the root of the zip file.) We should be able to open the archive, copy your files directly into the project, compile, and then run the project without doing anything else. Projects that do not compile or run will be marked zero.

4) Follow the naming convention exactly. Grading is done using script files so the project will not otherwise be found! If the project cannot be found due to a naming error, its grade will be zero.

Convention: ***lastName.firstName.labName.zip***

Example: **Coulton.Jonathan.PathPlannerLab.zip**

Lab Specialist Assistance Guidelines

The role of lab specialists in AFI is to facilitate student work and guide the learning process. However, students are expected to perform work independently, and more specifically, to write and debug their own code. Lab specialists are not to look at student code or debug student code errors, as these are important elements of the assignments themselves. Lab specialists will help students by clarifying concepts and algorithms, examining program output in order to help students track down program flaws, advise students on architectural elements of projects, and by engaging in other activities that help students to complete projects through their own work.

Helpful Hints

The following is a list of suggestions that may help you in lab:

Be sure you understand the concept before trying to write code applying it. If you don't get the concept, you will waste time and become frustrated. You have to complete ***and*** understand assignments to get full credit for them.

Try to ask conceptual questions. The lab instructors are in lab to help you understand concepts and make suggestions. Often a question will be followed by other questions that give hints about solutions that you can be found with a little searching. Lab instructors cannot correct mistakes line by line and remember: in the work force, you will need to solve your own bugs and problems.

When you encounter bugs, try to narrow down where things went wrong. Don't get frustrated. If you suspect you are not clear on the concept, ask us. Before asking for help with debugging, try to use a debugger to examine the variables and call stack. Improve your debugging skills now so you're ready for the Final Project (and real life!)

Always complete each lab. Learn the concept so you can do reasonably well on its corresponding quiz as well as the final exam (see the ***Due Dates*** for more information on dates and late penalties.)

If you get done early with your projects, work on your research project. Do not try to leave early if there are things to be done. Do not distract others in lab even though you may be already done with the project as they may be under pressure to complete it.

Be sure to save a copy of everything. Again: be sure to save a copy of everything! All of the lab projects are submitted electronically in digital format. If you believe you turned in a project that does not appear on the LMS and you do not have it, your project score will be a zero.

Don't micro-optimize! Get the code right, and then use the profiler. Micro-optimizations are bad for two reasons: they make your code much harder to debug and get correct, and they're also often wrong and will actually make your code slower. The only way you can justify optimizations is with data, and the only way to get data is by using the profiler (or using print statements).

Do it dumber (to start). AKA "I took out all this code and now it works better." Always implement the absolute simplest solution for a lab first. This will ensure that you fully understand the concepts involved and help avoid bugs. Not only that, but often the simple solution will outperform more complex solutions you may be considering, especially since you can iterate on it more during testing.