

## CS 461 – ARTIFICIAL INTELLIGENCE

### HOMEWORK #4 (5% OR 10 POINTS)

Assigned: **Wed 20 Nov 2019**

Due: **Wed 4 Dec 2019 \*\* 2 pm \*\***

---

*You can do this homework in groups of 5 (or less). Your group for this homework should normally coincide with your term project group. In any case, do not forget to indicate clearly the students who are submitting this homework (i.e., write at most five names on the submission).*

*You must submit your entire homework (include all the original code written) to our TAs. Just a single submission per group! **Our TAs will soon send you a note explaining the mechanics of submissions. They may also tell you whether there'll be a need to submit hardcopy, etc.***

*Any programming language can be used as long as you have it available on a portable computer. Needless to say, a group member should be prepared to give a homework demo (individually and using that portable computer) when requested to do so by our TAs.*

---

In this homework you are expected to obtain the class-precedence list for a given class hierarchy. The normal way of doing this is to implement the 'fish hook algorithm' given by Winston (chapter 9). But reuse of any topological sorting program is also permitted, provided that you clearly state where you found it. You must also tell, in precise terms, how you modified it to fit the needs of this homework.

You should test your program with the test examples (class hierarchies) in the following pages to make sure that it reports class-precedence lists correctly. The input to your program (i.e., the graph-theoretic data structure underlying a given hierarchy) should be manually prepared by you in order to faithfully capture it. The output of your program should simply be a linear list, with the highest and lowest precedence ends clearly identified. (In all test examples, feel free to add, for the sake of observing the convention of chapter 9, the 'universal' class called **Everything**.)

Do not forget to submit the entire code (even if it is not really yours).

Your program should have a simple control for 'single stepping' (tracing your code) so that you and the TAs can inspect the intermediate stages of the problem-solving process in an incremental fashion. Needless to say, this is also useful for debugging your program during the development stage.

---

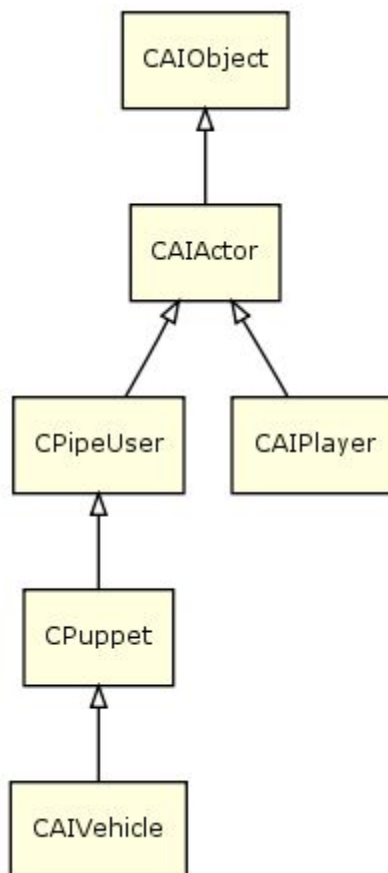
### GENERAL REMARKS (THESE ARE APPLICABLE TO ALL HOMEWORK ASSIGNMENTS)

- IF YOU ARE REQUESTED TO SUBMIT A HARDCOPY AT ANY TIME IN THIS COURSE, MAKE SURE THAT WHAT YOU SUBMIT IS CLEAN AND FULLY MACHINE-GENERATED. IF THERE IS A HANDWRITTEN ADDITION OR CORRECTION ON A PRINTOUT, YOU'LL DEFINITELY LOSE POINTS.
  - Late submissions will first have 2 points deducted categorically. Then they'll have 2 points deducted for every late day. (A new day begins at 12:01 midnight.)
- 

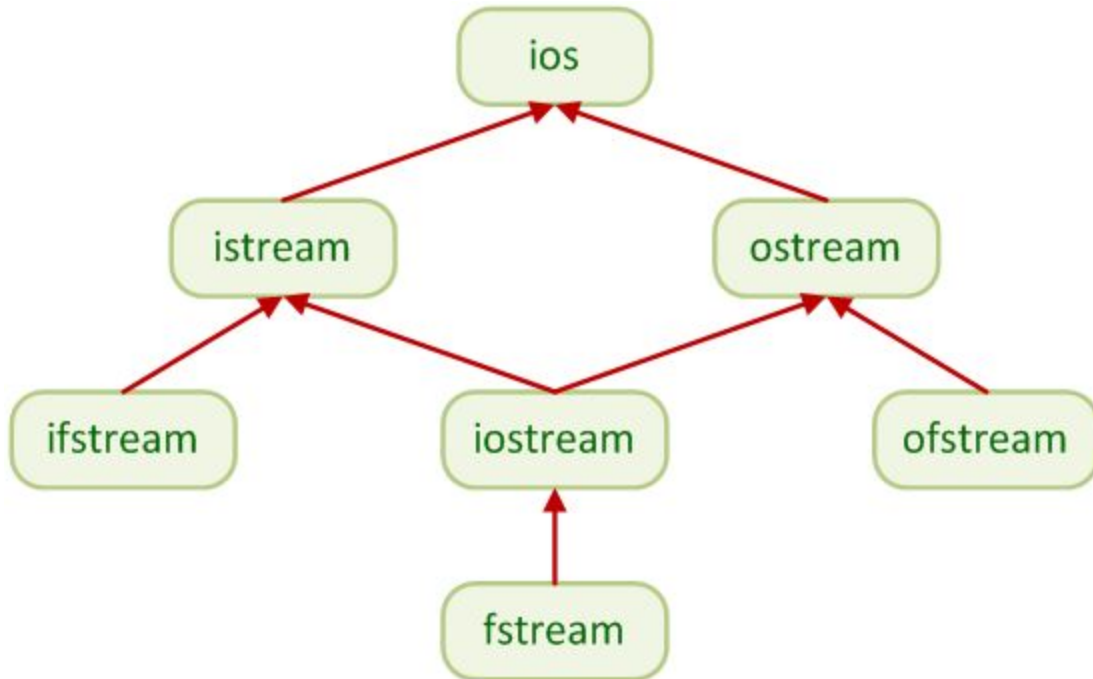
### TEST EXAMPLES AND GRADING

The first example is worth 2 points. The second example is worth 3 points. The third example is worth 3 points. The software you've written (or reused with modifications) is worth 2 points.

Example 1 (Compute 2 lists: for **CAIVehicle** and for **CAIPlayer**)



Example 2 (Compute 3 lists: for **ifstream**, for **fstream**, and for **ofstream**)



Example 3 (Compute 3 lists: for **Consultant Manager**, for **Director**, and for **Permanent Manager**)

