

## CS 461 – ARTIFICIAL INTELLIGENCE

### HOMEWORK #6 (6.5% OR 13 POINTS)

Assigned: Wed 14 April 2021

Due: Wed 28 April 2021 \*\* 2 pm \*\*

(DO NOT SUBMIT YOUR HOMEWORK BEFORE THIS DEADLINE; SEE BELOW)

---

*Submit your entire homework (the original code plus example runs) via Moodle, as specified by our TAs.*

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

---

In this last homework of the semester, you are expected to obtain the class-precedence list (CPL) for a given class hierarchy. To this end, you'll implement the 'fish hook algorithm' given by Winston (chapter 9; all page numbers in the sequel refer to this chapter). *Caveat: In light of the preceding sentence, reuse of existing topological sorting programs is strictly forbidden. Just implement Winston's aforementioned algorithm from scratch.*

A proper CPL obeys the two rules (single bullet points) given on p. 187 and p. 190.<sup>1</sup> You should test your program with the examples (class hierarchies) on the next page to make sure that it reports CPLs 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 the test examples, feel free to add, for the sake of observing the convention of chapter 9, the class called "Everything.")*

Your program should have a simple control for 'single stepping' (tracing your code) so that the TAs can inspect the intermediate stages of the problem-solving process in an incremental fashion. *Essentially, the (continuously updated) table of fish-hook pairs is what you need to show all the time.* Needless to say, this will also be useful for debugging your program during the development stage.

---

<sup>1</sup> N.B. There's also an important tie-breaker that is explained on p. 194. Do not forget to implement it.

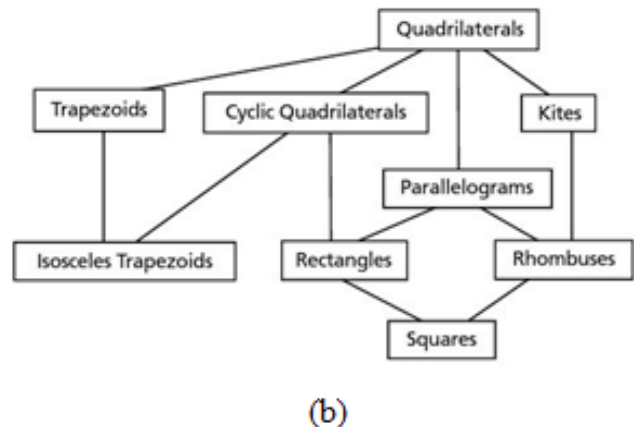
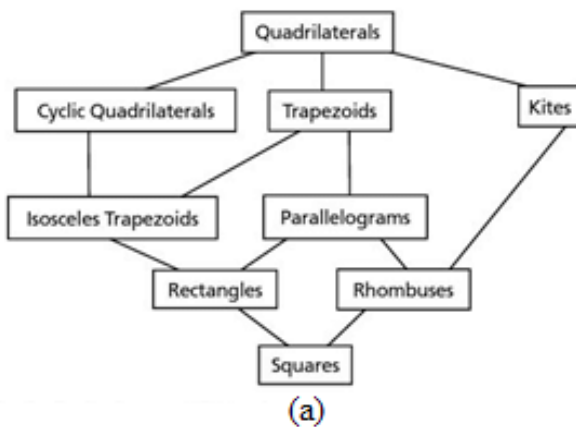
## TEST DATA AND GRADING

The third ('surprise') example below will be sent on the evening of Tue 27 April. (THUS, PLEASE DO NOT SUBMIT YOUR HOMEWORK UNTIL THEN.)

The quality of the code you'll write is worth 3 pts: 3 (good), 2 (average), 1 (poor).

**Example 1 (3 pts)** - See part (a) of the following figure. (You can think of all line segments as up-arrows with ako labels.) Compute a CPL for "Squares."

**Example 2 (3 pts)** - See part (b) of the following figure. (Again, you can think of all line segments as up-arrows with ako labels.) Compute 2 CPLs: one for "Isosceles Trapezoids" (1 pt) and one for "Squares" (2 pts).



**Example 3 (4 pts)** - Forthcoming...