

# CS-4701 Practicum in Artificial Intelligence

## Fall 2013

### Project Proposals

**Due Date:** Monday, October 21, midnight on CMS.

For your pdf file, use filename in following format:

**CS-4701-[1st\_netid]-[3\_key\_words].pdf**

1st\_netid is alphabetically first netid in team; key\_words can be the main words from the title.

**Format:** The proposal writeup (pdf) should contain six components:

1. **Title** of your project.
2. **Names** with Cornell netids of team members.
3. A clear, concise **description** of what you plan to do.
4. The general **approach** you'll use (e.g., heuristic search, game tree search, learning, inference rules, or neural networks).
5. An explicit, coherent **plan** for a quantitative and/or qualitative evaluation of your approach and system.
6. A **timeline** for your implementation and evaluation.

The proposal should not be more than about **two pages** in length. If you'd like a partner for the project and haven't found one, indicate this somewhere on the proposal.

Proposals will be graded Satisfactory/Unsatisfactory based on the clarity and believability of the above components.

Proposals can be modified for content in response to my comments/suggestions.

NOTE:

1. Email me at selman@cs.cornell.edu if you'd like some feedback on early ideas or are looking for team members. Use as subject: **CS-4701 [text]**
2. For projects ideas, see the CS-4701 web page. Check R&N for chapters related to your project.
3. For the **final evaluation**, I will have a brief meeting with each group. You will need to present your project with a **powerpoint (about 10 to 15 slides)** and a brief **demo of your software**. So, it is important to include a good **graphical interface** for the demo (if applicable). You will also need to hand in a **write-up** on the project (**between 10 and 20 pages**). The write-up should include a clear description of the overall goals of the project, the software written, and the results an evaluation of your system with **various observations on the AI components and their performance**. For the latter aspect, you should think about ways to study your system. For example, in a project on game playing, you should have your program play itself or variations of itself. You can evaluate how well various heuristics perform and the effect of increasing search depth. In a learning related project, you can consider the effect of training data size and the complexity of the learned concept or neural net. Don't hesitate to drop me an email if you have questions about these issues.