CS 4301: Declarative and Logic Programming
001 — Fall 2025

Instructor Information

Name: Ilan Buzzetti
Office: ECSS 3.612
Email: ixb140230 (@utdallas.edu)
Office Hours: TTh 1:30 – 2:30

Class Information

Dates: Tu Aug 26, 2025 – Tu Dec 9, 2025
Time: TThu 11:30 – 12:45
Classroom: ECSX X.XXX

Course Content

This class centers around the 2 projects which expect you, as the student, to creatively and independently apply the material in this class. We begin by learning Prolog, the most popular logic programming language, and how to think as a logic programmer. We quickly move on to demonstrating problems and applications for which logic programming is especially suited. This puts the fundamentals into context and starts exposing you to common and fun applications of logic programming.

After completing this course you will be a competent Prolog programmer and able to tackle realistic problems using a logic programming approach. This forms the basis of knowing when to apply logic programming to a problem, and when other approaches will be more fruitful.

Prerequisites: CS 3345.
Complementary Courses: CS 4337, CS 4349.

Grading

Quizzes	20%
Project 1	20%
Project 2	25%
Presentation	35%

Approach

This class will be project oriented. It is my hope that you come out of this class having accomplished something you are proud of and that you can show off on your resume, to your parents, to your dog, etc. I will release the project guidelines on the first day of class.

The daily quizzes are there as knowledge checks. They will test your comprehension of the material. This is helpful to 1) give you a sense of what you're not understanding without forcing you to do homework and 2) give me a sense of what you're not understanding without grading homework. Each quiz is worth 10 points and only the top 10 quiz grades will contribute.

In the final presentation you will have the opportunity to present one of your projects or another application/use case for logic programming that we have not discussed in class. If you do not see yourself as a public speaker, worry not! You need not be an excellent orator to get full points. The presentation grade is there to motivate you to document and polish your project well.

Course Objectives

After this course, you should be able to...

- Think of solutions to problems in terms of rules.
- Translate natural language problems into formal specifications of correctness into logic programs that solve them.
- Reason about recursive relations.
- Feel comfortable writing prolog code and navigating the online documentation.

Textbooks

Textbook: *The Art of Prolog*, by Leon S. Sterling and Ehud Y. Shapiro.
Textbook: *The Power of Prolog*, by Markus Triska.

We will be closely following the *The Art of Prolog* for learning the fundamentals. We will focus on the practical aspects of programming. You do not need to learn the vocabulary nor theoretical basis that is treated in the first part of the book. A digital version of the book is made freely available at the publisher's website [link].

The second part of the course will feature a selection of material from *The Power of Prolog* which is made freely available by the author at https://www.metalevel.at/prolog.

Phone and Device Policies

All electronic devices should be silenced, put away and left to their own devices during lecture unless otherwise specified. Especially during the first part of the course when we aim to develop a clear mental picture of what is happening when a logic program executes. When we start delving into more specialized areas having a laptop to test and clarify your own questions may be useful. If electronics become a hindrance more than a help they will be disallowed entirely.

Class Attendance

As per department policy more, three consecutive absences results in a letter grade reduction from your final grade, four consecutive absences results in a failure. If you are dismissed from lecture due to problems during the lecture, e.g. disruptive behavior or unauthorized cell phone use, then this dismissal will be recorded as an absence.

Exams

No Exams. Rejoice! And do some creative work, hopefully on your projects.

:^)

Use of Student Work

In compliance with the federal Family Educational Rights and Privacy Act, registration in this class is understood as permission for assignments prepared for this class to be used anonymously in the future for educational purposes.

Respect Policy
I respect your time:

- I will come prepared to help you understand the course material and prepare you for quizzes/exams.
- Communication is key: I cannot help you if I do not know what is going on.
- I am here to help you, this is your time, so let me know what I can do to help you succeed.
- If there is something that you would like me to do differently, please, let me know. I am happy to work with you to make class the best it can be.

Respect my time:

- Be on time to class.
- Pay attention when I am talking to you.
- Come to class prepared by doing the work and going to office hours when you need help.

Respect each other:

- Do not be disruptive. If you need to take a call or text someone, take it outside.
- Work with each other to find solutions. You learn more by helping each other.
- Allow one another to make mistakes. This is an important part of the learning process.
- Use respectful language when talking with one another.

Tips for Success

- Be proactive about your success in the course.
- Do not procrastinate! Begin your assignments and studying early!
- Attend every class and recitation.
- Ask questions whether it is during class, recitation, office hours, at the math clinic or via email to your instructor.
- Form a study group! Working together will help you and others better understand the course material as you can work through different difficulties and offer each other clarifications on concepts.
- Do problems! Reading through your notes is not enough. Seek out new problems and work through them carefully. When you are done, check your answer. If you are wrong, examine carefully what misunderstanding occurred and how to avoid it in the future. If you were correct, examine if there was a faster way, check to see if your solution 'flowed' and was easy to read, and think over what concepts/computations were used and what 'type' of problem the exercise was.
- Always check to be sure that you understand when a statistical computation can be used and possible sources of error or bias in the statistics computed.
- Every time you approach a new concept, carefully think how it could be applied in your own field of study.
- Carefully check your code when you use any statistical computation device.