JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Computer Science EN.601.432 / EN.601.632
Natural Language Processing
Spring 2026 (3 credits)

Instructor

Professor Jason Eisner (he/him) <jason@cs.jhu.edu>
Office hours: After class in the classroom (4–4:30), or by appointment
Office location: Hackerman 324C, or https://wse.zoom.us/my/jeisner1

Teaching Assistant

TA: Jacopo Teneggi <jtenegg1@jhu.edu>
Office hours: TBA. See the class homepage, http://cs.jhu.edu/~jason/432.

You can reach all the course staff at once via a private post on Piazza, or via cs432-staff (at) cs.jhu.edu.

Meetings

- **Lectures**: Mon/Wed/Fri 3–4:15 pm, Hodson 213
- **Recitations (optional problem-solving sessions)**: TBA
- **Online discussion**: https://piazza.com/jhu/spring2026/601432

This class is in the "flexible time slot" MWF 3–4:30. Please keep the entire slot open. Class will usually end at 4, followed by office hours in the classroom from 4–4:30 (stick around to get your money's worth). However, class will sometimes run till 4:15 in order to keep up with the syllabus. I'll try to give advance notice of these "long classes," which among other things make up for no-class days when I'm out of town.

We'll also schedule a once-per-week discussion session led by your TA. This optional session will focus on solving problems together. That's meant as an efficient and cooperative way to study for an hour: it reinforces the past week's class material without adding to your homework load. Also, if you come to discussion session as recommended, then you won't be startled by the exam style. (The discussion problems are taken from past exams and attempt to be interesting, which makes them challenging.)

If you miss a lecture, please email the professor to get access to a video recording. Watch it promptly so you can follow subsequent lectures. Lecture recordings are also available for review upon request, and will be made available before each exam.

If you click the CC button, automatically generated closed captions will show up on the video. You can also view the full transcript and use it to navigate to a specific part of the video. (The video lessons have manual captions. The lecture recordings have automatic captions; these aren't perfect, but they might help you understand my words better.)

Recitations will generally not be recorded, but the materials from recitation will be posted on the class website.

Resources

Everything you need will be linked from the class homepage, http://cs.jhu.edu/~jason/432.

Go there now! Make sure to sign up for the Piazza site, and make sure to watch lecture videos when they are assigned later. (And of course, come to class, come to recitations, do the homeworks, take the exams.)

There's no textbook, as I don't know of any other courses like this one. Supplementary readings will be linked from the course homepage, e.g.,

- Programming Pearls: Little Languages (Bentley, 1986)
- *Constraint Processing* (Dechter, 2005)
- ECLiPSe—A Tutorial Introduction (Cheadle et al., 2017)
- Prolog Programming: A First Course (Brna, 1988)
- ZIMPL User Guide (Koch, 2015)
- Dyna: Extending Datalog for Modern AI (Eisner and Filardo, 2011)

Course Description

Catalog description: Suppose you could simply write down a description of your problem, and let the computer figure out how to solve it. What notation could you use? What strategy should the computer then use? In this survey class, you'll learn to recognize when your problem is an instance of satisfiability, constraint programming, logic programming, dynamic programming, or mathematical programming (e.g., integer linear programming). For each of these related paradigms, you'll learn to reformulate hard problems in the required notation and apply off-the-shelf software that can solve any problem in that notation – including NP-complete problems and many of the problems you'll see in other courses and in the real world. You'll also gain some understanding of the general-purpose algorithms that power the software.

Remarks: You could regard this as an alternative programming course. A programming course teaches you how to use a programming language to solve problems. It also outlines how your computer will actually execute the code you write in that programming language.

The languages we'll be using aren't conventional programming languages. They are powerful problem description languages that focus on special-purpose computation. These languages are declarative. That is, you use them to specify a problem, not a solution. But they are backed up by solvers that do a good job of finding the solution efficiently in most cases. This course will survey some declarative languages and examine the kinds of solvers that people have written for them.

How do these languages relate to conventional ones? Conventional languages help you build arbitrary large systems in a modular way. You can think of a solver as a particular powerful module that handles many problems of a particular sort. To explain your particular problem to the solver, you use a declarative language.

Course Goals

Specific outcomes for this course:

- Understand the properties of various declarative modeling languages, including standard reductions among encodings.
- Understand how to formulate novel constraint satisfaction or constrained optimization problems in various declarative modeling languages.
- Acquire experience at solving constraint satisfaction or combinatorial optimization problems in practice, e.g., by writing a program that encodes the problem in a modeling language, runs a modern solver, and decodes the result.
- Understand the generic algorithms that modern solvers use to execute declaratively specified problems, including an appreciation of how runtime will vary across problems and across solution strategies.

In the end, I hope that Declarative Methods, like all good courses, stretches your mind and leaves you with new ways of thinking. Not to mention all this.

Course Topics

- Introduction to little languages
- SAT: The SAT problem, applications of SAT, and algorithms for deciding SAT. (Programming: SAT solvers)
- Constraint satisfaction problems (CSPs) : Problem definition, representing problems as a CSP, and algorithms for solving CSPs. (Programming: ECLiPSe language and solver)
- Prolog: Introduction (syntax and semantics), problem (e.g. data structure) representation in Prolog, how methods used by the Prolog solver. (Programming: Prolog and ECLiPSe language and solver)
- Mathematical Programming: mathematical model problem formulation, types of mathematical models, algorithms for finding optimal solutions, encoding complex problems as a set of constraints. (Programming: ZIMPL/SCIP)
- Dynamic Programming: background, recognizing a DP solution, procedural implementation for a solver. (Programming: Dyna)

The topics may change during the semester. See the class website for current details: `http://cs.jhu.edu/~jason/432`.

Prerequisites

- Data Structures (601.226)

Course Expectations & Grading

Your work will probably be weighted as follows:

- **35% homeworks**, equally weighted (see lateness policy)
- **10% midterm exam**
- **25% final exam**
- **25% term project**
- $\approx$ **5% participation**

Participation includes your contributions to an interesting and useful class discussion, whether synchronously during class sessions, or via public posts or replies on our Piazza discussion website. This includes asking questions, of course. We've found that participation has a substantial effect on the final grade.

Homeworks will be submitted via Gradescope. We expect to have 5 homeworks, which focus on *using* the different declarative languages.

We plan to have two exams—a midterm exam and a final exam. These will test both how to use the languages and also how the solvers work inside. They will have some challenging questions. In the recitations, you'll get to work in small groups to solve interesting problems from past exams.

Since some questions are hard or are graded to maximize variance, we use a curve to pull the grades back up. If your grade was already high in absolute terms, it will stay high (so if everyone gets 100, everyone gets an A+, not a C).

Extra credit is added only after the curve is determined. Thus, extra credit is not required to do well, although it can help make up for low grades elsewhere.

Late Homework Policy

Since emergencies sometimes arise, I'll grant extensions on request, in multiples of 24 hours, up to a *total* of up to 10 "late days" during the term. The full policy and advice on how to use it are at http://cs.jhu.edu/~jason/432/late-policy.html.

Audit Policy

If you register for the course as an auditor (or switch to this status), please let me know that you are auditing. You're still expected to attend the lectures and to participate in them like any other student. (That's what it means for the course to appear on your transcript with an AU grade.) If you are not able to sustain this level of commitment, you should drop the class.

You are also welcome to do any homeworks or exams that are useful to your learning. We will grade them and give you the usual feedback.

Key Dates

Please see the class website: http://cs.jhu.edu/~jason/432.

Assignments & Readings

Please see the class website: http://cs.jhu.edu/~jason/432.

Assistance and Collaboration Policy

Understanding course content: We really want you to understand the course material! Pretty much anything is fair game for this. We already provide lectures, reading handouts, recitation puzzles, office hours, and optional recommended readings. But you might also might look for other readings, talk to classmates or CAs in person or on Piazza, or discuss in depth with an AI chatbot.

Please share useful materials and insights on Piazza so that others can also benefit. We are especially curious about whether discussion with AI chatbots can help your understanding. Note that ChatGPT offers a "Study Mode" that is designed for this (direct link).

Doing graded work: The point of the homeworks and exams is for you to wrestle with problems. That's how you learn, and also how you confirm your understanding. It's also how we confirm your understanding (for grading). We want you to succeed! However, if you don't solve the problems yourself, then you're not the one who is succeeding. So, it's ok to get help, but not in a way that solves the problems for you. Here are key principles:

- It's always fine to get help from the course staff. They will use good judgment about how much help is appropriate.
- Some homeworks allow collaboration with a partner. (In most cases, this means making a single group submission.) You are expected to do the work *together* with your partners, not divide it up: if you didn't work on a question, you don't deserve credit for it! Your solutions should emerge from collaborative discussions with the *whole group* participating.
- If you get help in your graded work from anyone else–that is, help beyond course staff and your partners—then you need to disclose this in detail on your homework submission. This includes help from
  - other humans
  - AI chatbots
  - documents that give the answers
- You only need to disclose help on actually solving the graded problems. You don't need to disclose help that merely gets you up to speed so that you are prepared to do the problems, such as
  - "how do I read and write JSON files in Python?"
  - "why isn't matrix multiplication commutative?"

  - "give me 5 matrix expressions in PyTorch and their mathematical notation equivalents in TEX"
  - "I'm suddenly confused about lecture 12"
- You need to do substantive work to get credit:
  - Even if you get some help, **your written solutions should always be in your own words**. This ensures that you actually have the ideas in your head. Your English will be good enough for us to follow; there's no need to have an AI rewrite it for you.
  - **You should also write your own code**; don't use an AI-powered code editor. This ensures that you are competent to write or review the core code for NLP algorithms. (If you're thinking that it might be appropriate to have AI fill in non-core code—boring parts that have little to do with NLP—there's actually no need for that in this course. We have already filled those parts in for you, by providing well-crafted "starter code" that you should read.)
  - Don't provide an AI with the actual homework questions or starter code. That's too close to having it solve the problems for you.
- If you're not sure what's allowed, or you'd like permission to do something that seems reasonable to you, please ask us! And err on the side of disclosure. Better safe than sorry.

The material below is standard for CS Department syllabi, except for portions in this color.

Integrity

The strength of the university depends on academic and personal integrity. You must be honest and truthful in this course, abiding by the CS department's Academic Integrity Code:

> Cheating is wrong. Cheating hurts our community by undermining academic integrity, creating mistrust, and fostering unfair competition. The university will punish cheaters with failure on an assignment, failure in a course, permanent transcript notation, suspension, and/or expulsion. Offenses may be reported to medical, law or other professional or graduate schools when a cheater applies.
>
> Violations can include cheating on exams, plagiarism, reuse of assignments without permission, improper use of the Internet and electronic devices, unauthorized collaboration, alteration of graded assignments, forgery and falsification, lying, facilitating academic dishonesty, and unfair competition. Ignorance of these rules is not an excuse.
>
> Academic honesty is required in all work you submit to be graded. Except where the instructor specifies group work, you must solve all homework and programming assignments without the help of others. For example, you must not look at anyone else's solutions (including program code) to your homework problems. However, you may discuss assignment specifications (not solutions) with others to be sure you understand what is required by the assignment.
>
> ***If*** your instructor permits using fragments of source code from outside sources, such as your textbook or on-line resources, you must properly cite the source. Not citing it constitutes plagiarism. Similarly, your group projects must list everyone who participated.
>
> Falsifying program output or results is prohibited.
>
> Your instructor is free to override parts of this policy for particular assignments. To protect yourself: (1) Ask the instructor if you are not sure what is permissible. (2) Seek help from the instructor, TA or CAs, as you are always encouraged to do, rather than from other students. (3) Cite any questionable sources of help you may have received.
>
> On every exam, you will sign the following pledge: "I agree to complete this exam without unauthorized assistance from any person, materials or device. [Signed and dated]".

> Your course instructors will let you know where to find copies of old exams, if they are available.

In this class (NLP), practice exam problems will be provided for you. You should not otherwise make use of homeworks or exams from previous years.

Report any violations you witness to the instructor. You can also contact:

- For undergraduates:
- the Office of the Dean of Student Life at 410-516-8208 or <studentconduct@jhu.edu>
- For KSAS Graduate Students: <rseitz5@jh.edu>
- For WSE Graduate Students: <christinekavanagh@jhu.edu>

Students with Disabilities – Accommodations and Accessibility

Johns Hopkins University is committed to providing welcoming, equitable, and accessible educational experiences for all students. If disability accommodations are needed for this course, students should request accommodations through Student Disability Services (SDS) as early as possible to provide time for effective communication and arrangements.

For further information about this process, please refer to the SDS Website or email SDS Homewood: <studentdisabilityservices@jhu.edu>.

Classroom Climate

The following statement was provided by the Homewood Council on Inclusive Excellence and is incorporated into all WSE course syllabi.

> Johns Hopkins University is committed to creating a classroom environment that values the diversity of experiences and perspectives that all students bring. Everyone here has the right to be treated with dignity and respect. Fostering an inclusive climate is important because research and experience show that students who interact with peers who are different from themselves learn new things and experience tangible educational outcomes. Please participate in creating a welcoming and vibrant classroom climate. Note that you should expect to be challenged intellectually by the instructor, the TAs, and your peers, and at times this may feel uncomfortable. Indeed, it can be helpful to be pushed sometimes in order to learn and grow. But at no time in this learning process should someone be singled out or treated unequally on the basis of any seen or unseen part of their identity.
>
> If you ever have concerns in this course about harassment, discrimination, or any unequal treatment, or if you seek accommodations or resources, please reach out to your instructor or the TAs, who will take your communication seriously and seek mutually acceptable resolutions and accommodations. Reporting will never impact your course grade. You may also share concerns with the Department Head (Randal Burns, <randal@cs.jhu.edu>), the Director of Undergraduate Studies (Joanne Selinski, <joanne@cs.jhu.edu>), the Director of Graduate Studies (Scott Smith, <scott@cs.jhu.edu>), the WSE Associate Dean of Outreach and Belonging (Darlene Saporu, <dsaporu@jhu.edu>), the KSAS Assistant Dean for Diversity and Inclusion (Araceli Frias, <afrias3@jhu.edu>), or the Office of Institutional Equity (<oie@jhu.edu>).
>
> In handling reports, people will protect your privacy as much as possible, but faculty and staff are required to officially report information for some cases (e.g., sexual harassment).

Personal Wellbeing

Illness. If you are sick, in particular with an illness that may be contagious, please notify me by email but do not come to class. I will provide you with a live Zoom link and/or a video recording of the lecture. We can also make arrangements to take exams remotely if necessary.

You may also contact Student Outreach and Support at <studentoutreach@jhu.edu> for help managing your coursework during an illness, with medical leave of absence being an option for a longer illness. For your medical needs, see the Student Health and Wellness Center.

Other Reasons to Miss Class. If you have to miss class for another reason, such as a religious holiday or a visa problem, please notify me by email in advance. I will try to accommodate you as above (within reason). In general, of course, you are expected to come to class.

Mental Health. JHU has several resources to support students. Many students struggle at times with stress, anxiety, and depression. The Counseling Center has many resources available to you:

- Johns Hopkins University Student Well-Being

In addition, The Johns Hopkins University Behavioral Health Crisis Support Team (BHCST) pairs experienced, compassionate crisis clinicians with specially trained public safety officers on every shift on and around the Homewood campus, seven days a week. The BHCST will provide immediate assistance to those who need it and, just as importantly, link individuals in crisis to ongoing support services in the days and weeks that follow. Call Public Safety, 410-516-5600, and ask for a BHCST clinician.

If you have concerns about a specific student, please contact:

- For emergencies (threat to self or others): 410-516-4600 or 911
- For on-scene mental health support: BHCST at 410-516-4600
- For undergraduates: Student Outreach & Support at 410-516-7857 or <studentoutreach@jhu.edu>
- For KSAS Graduate Students: Renee Eastwood, Assistant Dean for Graduate and Postdoctoral Academic and Student Affairs
- For WSE Graduate Students: Megan Barrett, Assistant Dean for Engineering Student Affairs

Family Accommodations Policy. You are welcome to bring a family member to class on occasional days when your responsibilities require it (for example, if emergency childcare is unavailable, or for health needs of a relative). Please be sensitive to the classroom environment, and if your family member becomes uncomfortably disruptive, you may leave the classroom and return as needed.