Name and Contact Information of Instructor:

Prof. Dennis Shasha  
Department of Computer Science  
Courant Institute of Mathematical Sciences  
New York University  
251 Mercer Street  
New York, N.Y. 10012-1110  
U.S.A.  
Email: shasha@cs.nyu.edu  
Web: http://cs.nyu.edu/cs/faculty/shasha/index.html

Brief Course Description:

Many problems in science, business, and politics require heuristics -- problem solving techniques that often work well but give imperfect guarantees. This course teaches heuristics as they apply to the design of scientific experiments, the resolution of economic or political negotiations, and in the construction of engineering devices in hostile environments. Students will work in small teams that will solve puzzles, conduct experiments, and build strategies for a competitive auction game. Students will use and learn computational tools, building on some programming experience in Python or other languages. The intent is to make you better able to face complex problems in any field you choose.

The writing center offers the support of writing tutors. You will be seeing a lot of them. They will help you with drafts, but will not grade you. The writing mentor leader is Leah Souffrant ljs17@nyu.edu.

Learning Outcomes:

Design approaches to problems that are quantitative, procedural, and adaptive, even when the problems suffer from incomplete information or are only approximately described.
Understand heuristic techniques from computer science, biology, and human history.

Design social, physical, or computational experiments that apply heuristic techniques.

Teaching and Learning Methodologies:

Lectures, readings, programming, and social/cultural experiments.

Methods and Dates of Assessments:

Class participation 5%, presentations 10%, and quiz work 15% (participation evaluated based on interaction with other presenters, responses to questions from the professor, and quality of student’s own presentation), three projects worth 20%, 20% and 30%. Each presentation of a book chapter should be accompanied by a two to three page writeup explaining some element that goes beyond what you find in the chapter you are discussing. Presentations of projects may be longer. Your writeup should go to the writing tutors within two days of the presentation and is due two weeks later. The experimental project (project 2 below) should be accompanied by an eight 10 page paper explaining the goals, the theoretical background, the results, and the conclusions along with a statistical analysis. There will be four or so short quizzes that may or may not be announced in advance.

Project 1: master one puzzle-based game, design a strategy that is likely to win and be prepared to take on challengers. (due on week 4, with draft to the writing tutors two days later and final to me by week 6).

Project 2: design and demonstrate an experiment to test problem-solving techniques and write a description of the goals, techniques, and conclusions (due on week 9). Draft to the writing tutors a few days later and paper due on week 11.

Project 3: compete in the design of methods for a competitive auction (due on week 13). Presentation but no writeup.
Course texts:

1. *The Prism and the Pendulum*
   Robert Crease
   Publisher: Random House Trade Paperbacks (October 12, 2004)
   Language: English
   ISBN-10: 0812970624

2. *Puzzles for Programmers and Pros*
   Dennis Shasha
   Publisher: Wrox; 1 edition (May 7, 2007)
   Language: English
   ISBN-10: 0470121688

3. *Natural Computing*
   Dennis Shasha and Cathy Lazere
   Paperback: 288 pages
   Publisher: W. W. Norton & Company (May 17, 2010)
   Language: English
   ISBN-10: 0393336832

4. *Statistics is Easy*, second edition
   Dennis Shasha and Manda Wilson
   Publisher: Morgan & Claypool [can be obtained online]
   Language: English
   ISBN-10: 1598297775

5. *Thinking, Fast and Slow*
   by Daniel Kahneman
   Farrar, Straus and Giroux; Reprint edition (April 2, 2013)
   Language: English
   ISBN-10: 0374533555
Course topics and Contents:

Section 1: Thinking quantitatively -- estimation in the natural and human-constructed world (e.g. how many MacDonalds in country X). Heuristic inference of causes in fast and slow styles of human thought. Thinking heuristically -- use of puzzles to approach difficult-to-solve and possibly adversarial problems systematically even under incomplete information. Start learning python through graduated examples.


Week 4 project: Be able to be the expert on some puzzle game and see if someone else can beat you at it. four page writeup on your strategy. Please submit your draft for the writeup two days after the competition to the writing tutors with final writeup due at week 6.

Week 1 (September 11, 2017): Motivation: when do we need heuristics? For unknowable information, incomplete information, situations where improvisation is inevitable. Estimation and adaptation examples. Students group themselves into pairs. Introduction to puzzle games on drecco site. Discussion of recognition heuristic. Groups decide which puzzles to be expert at. Install python, introduction to the shell, types and hello world.

Week 3 (September 25, 2017) Part II of Thinking book. Notions of probability: Monte Hall Puzzle, Lucky roulette, Medical Test problem Students state which game they expect to master. Python arithmetic and conditionals and loops.

Week 4 (October 2, 2017): Appendix of Thinking Book. First contest. Strategic thinking puzzles: social games, optimizing the worst case, check amounts. Squash club membership. Coach’s dilemma. Python functions along with recursion. Ten questions problem. Python file treatment and parsing. Submit 4 page writeup of your strategy to the writing tutors (each of you is responsible for two pages) by October 5. Final writeup due to me on October 17.

Section 2: Approaches to the design of great experiments. How does one construct a model of some target question well enough so that carrying out the experiment is fairly simple (e.g. that electrons move through a conductor and not protons)? We look at problems in physics, e.g. measuring the speed of electrons in metal. In parallel, we will teach more elements of the Python programming language including elements of interprocess communication. You will also design your experiment. You will see which statistically valid conclusions you can draw and provide a eight page writeup (four pages being the responsibility of each student in each two-student group). In addition you will as groups make a presentation about a chapter in the Prism and the Pendulum or Natural Computing book. The presentation should be accompanied by a two page writeup jointly authored.

Primary readings: 1. The Prism and the Pendulum 2. Statistics is Easy

Week 8 project: Students will construct their own experiment and demonstrate in class.

Week 5 (October 16, 2017): Basic notions of statistics using a resampling approach. Appendix of Thinking Book (if not done). Student presentation of Crease chapters one and two. Merge evens only. Chapters 1 and 2 of Statistics is Easy

Week 6 (October 23, 2017): Student presentation of Newton’s decompostion of light. Student presentation of Foucault’s pendulum. Students decide on their own experiment and discuss with me. Python interprocess communication.

Week 8 (November 6, 2017): Student presentations from Crease Cavendish and Rutherford chapters. Python interactive website

Week 9 (November 13, 2017) Student presentations of their own experiment with videos in class. Game-playing techniques. Python interactive wordsnake game given common list of words. Presentation of work of pioneers of computer science. A draft of student report on experiment (objectives, design, first videos and discussion of statistical approach that will be used and two sets of conclusions – one if the null hypothesis is reject and one if it isn’t) to the writing tutors by November 16, 2017 with a final writeup to me on November 28.

Section 3: Approaches to the design of engineered devices that must face unpredictable hazards. Biological notions of feedback and evolution can contribute to a new form of adaptive and robust design. The discussion will center on non-examples like the Bhopal chemical plant as well as examples like marketplaces like the date and nut center near campus. The final project is an auction game in which items of different types will appear in an unknown order and your job is to obtain three of the same type. We will conduct several auctions of this type. You will describe your strategy in a presentation, but without a writeup.

Primary reading: Natural Computing

Week 13 Project: Student teams will build a program to “win” a series of auctions by being the first team to acquire three items of the same type.

Week 10 (November 20, 2017): Student presentations of (i) Rodney Brooks and (ii) Glenn Reeves/Adrian Stoica. Python work on final project, basic algorithm and plumbing and random algorithm.

Week 11 (November 27, 2017) 8 page report on experiment due: Student presentations of (i) Louis Qualls and (ii) Amrut/Jake. Python final project preparation.
Week 12 (December 4, 2017): Student presentations of (i) Nancy Leveson and (ii) D. E. Shaw. Student presentations of whichever other chapter you like. Python final project preparation.

Week 13 (December 11, 2017): Project 3: Auction Game. Students compete and then present their strategies using presentation software, but no formal writeup and no need to go to the writing tutors.

Week 14 (December 12, 2017): Wrap-up of any unfinished business.

Relationship of this course to others: useful to students in the social, natural, computational or engineering sciences. Not specifically redundant to any other course.

Comments from students from 2016:

Professor Shasha was always very accessible and would answer questions students had in an email that would cc the entire class. The brevity in the description of assignments allowed for great creativity in program design and made us truly and deeply learn how to code, but at the same time the class could become quite disorganized. The schedule did not adhere to the syllabus, but there was still a vague structure and a routine in the seminar.

Shasha was incredible. One of the best teachers I’ve ever had, and he made me much more enthusiastic about computer science. I loved everything from the content to the assignments to the subjects of lectures in the class. It could be a little bit more organized, but overall was excellent.

Professor Shasha thinks we are as capable as students on any level. He challenged us, but also was understanding if we couldn't grasp something. I love that he puts faith in his students and values their education over the grades they receive. Wonderful teacher. To improve the course I think it would be interesting to go deeper into the various problem solving techniques we learned in class. We learned about many heuristics, but I feel that in my programming I didn't always implement techniques we learned in class into my code. In some assignments it was hard to translate what we learned in class into a more efficient program.

Undergraduate Writing Tutors Program:

In this class, we are fortunate to have help from the Undergraduate Writing Tutors Program. Writing tutors are curious, well-trained peers who provide feedback to students on drafts of writing assignments. Their role is to encourage and challenge students to strengthen their writing and clarify their ideas. Writing tutors are trained to support the aims of the class, learning about the expectations for writing in the class and listening and responding carefully to individual student writers. While writing tutors are not Teaching Assistants and will not assess papers, they will focus writing conferences on questions that generate clearer writing and stronger thinking.
about the content. Writing tutors will also look for patterns of grammatical error in student papers, explaining how students can learn to correct these errors. The writing tutors’ main goals are to help students develop their writing and thinking in response to particular assignments and to become better writers over the long term.

Writing tutors take a semester-long practicum to learn to think more deeply about writing and to develop practices for working with peers on writing during individualized conferences. Tutors audit several classes or recitations and read some course materials in the classes where they tutor. Their primary aim is, however, to work with students through a practice-based approach to writing and revising. That is, they will ask questions and work to prompt students to reread, rethink, revise, and craft new writing during conferences.

Students are required to participate in the program for each designated paper assignment, submitting a draft of their paper on time for written feedback and attending a scheduled, 30-minute long, one-on-one conference. Writing tutors should receive complete drafts from students, not outlines or rough notes. Late submission of drafts to tutors and missed conferences are reported to the Professor, who may reduce a student’s final grade as a consequence.