ACM Data Science Task Force Course Example

CS 35: CS for Insight
and CS 181y: CS for Inquiry

Harvey Mudd College (Claremont, CA)
Zachary Dodds
dodds@cs.hmc.edu

Link to all materials: https://www.cs.hmc.edu/~dodds/cs35/

Knowledge Areas that contain competencies (knowledge, skills, and dispositions) covered in the course

Knowledge Area	**Total number of Contact Hours (50% lectures, 50% lab based)**
AP	**14**
ML	14
CCF	14
PDA	14
AI	28

Where does the course fit in your undergraduate Data Science curriculum?

This is CS2 and DS2 for non-majors. CS1 is required; traditional CS2 courses "implement data structures"; this course takes the path of "With only CS1, you can stand on the shoulders of 10,000,000 human hours available in Python's libraries!"

Is this course from or used in other curricula/majors?

Yes and no. It was designed because ambitious non-CS and non-DS students knew they needed more CS, but CS2 is only *directly* useful for pathways overlapping CS in identity. (Indirectly, all liberal arts pursuits are useful, so – yes – CS2 is useful for all paths *as a liberal art*!) What are those ambitious students looking for in terms of specific skills? (The conceptual underpinnings can be taken for granted: they're in both.) They seek to leveraging computing, CS and DS alike, for insight *in other fields*!

What is covered in the course?

CS for Insight seeks to build upon CS1's mindset and skill set in order that students emerge feeling justifiably and demonstrably confident that they can tackle the following challenges:

- (1) extracting the computational essence of non-CS problems (or correctly declaring that a problem does not have a computational component)
- (2) proactively assembling -- from existing libraries, tools, and datasets, atop CS5's foundational skills -- a software system which sheds light on a problem beyond CS itself; this student-learning objective would include each student's creation of an online portfolio of their work in the class
- (3) analyzing the output and design of such software in light of (a) its algorithmic and data-structure efficiency, (b) its human-development and -maintenance efficiency, and (c) its ability to provide insight for the problem(s) at hand - and suggest productive new directions for research
- (4) practice presenting the rationale, design, and results of the three skills above

What is the format of the course?

Traditional full course: 90-120 minutes of lecture; 90-120 minutes of lab; assignments.

How are students assessed?

Weekly assignments, noted above, and a self-designed final project (with some milestones + other guidance), including a final presentation to the whole group.

Course tools and materials

Python! For our interfaces, we use a mix of VSCode + command-line / Jupyter Notebook / hybrids of the previous two (VSCode's "Python Interactive") All freely available. We encourage Kite, IDE extensions, and other tokens of authentic *personal* engagement with *artificial* thought.

Why do you teach the course this way?

CS is undergoing a change in its fundamental identity. Yes, there will always be a need for software engineers, and the field will always enthusiastically prepare students who choose that path. Yet with each year, software engineering recedes in relative importance for CS.

Not replacing but eclipsing the traditional "CS for Software" mindset are the opportunities of "CS for Insight," i.e., computing with the purpose of providing insight to other endeavors -- specifically, insight that would be impossible or difficult to obtain without the tools of modern machines, their interfaces, and their libraries.

These insights address questions that software engineers would not ask -- because they are not software-engineering - or even CS - questions. CS 35 seeks to build the skillset, knowledgebase, and confidence/comfort with asking and answering those kinds of computational questions.

Body of Knowledge coverage

KA	Sub-domain	Competencies Covered	Hours (adds up to 42)
AP		Recognize and apply the main approaches and techniques that support the better presentation of information	7
ML	Classification	Select giving justification an approach to classification from kNN, decision trees, RFs, Neural Nets (all in SciKitlearn)	7
CCF		Select and apply appropriate architectures and tools for particular tasks	7
PDA		Select and utilize appropriate algorithms and data structures in developing software	7

AI	Computer Vision	No longer considered a subset of AI. Competencies: Digest computer vision's challenge by scripting several task-solutions	7
AI	Natural Language	No longer considered a subset of AI. Competencies: Itemize recent developments in NLP by using recent libraries	7

Additional topics
Notable topics covered in the course that you do not find in the ACM Draft 2 Computing Competencies for Undergraduate Data Science:

In addition to data-science's core, this course deliberately expands students' comfort and confidence generating data and "tidying" up sources of data too human to process as-is. That is, this course requires neither a data-science identity as input mindset, nor as output mindset. In fact, its goal is bringing computer-science and data-science problem-solving to other academic identities.

Recent update. These skills turn out to be useful for Computer Science and Data Science majors, as well. In 2021, our institution has created a major-elective version of this course, numbered CS181y. It differs in intensity, but not intent.