



I'm Afraid I Can't Do That, Dave:

Challenges and Opportunities for Teaching Programming

Daniel Spikol - 15 Juni 2024 - DIKU

Intro

DAVE: Open the pod bay doors, HAL.

HAL: I'm sorry, Dave. I'm afraid I can't do that.

DAVE: What's the problem?

HAL: I think you know what the problem is just as well as I do.

DAVE: What are you talking about, HAL?

HAL: This mission is too important for me to allow you to jeopardize it.

DAVE: I don't know what you're talking about, HAL.

2001: A Space Odyssey (1968), written by Stanley Kubrick and Arthur C. Clarke,
uncredited story "The Sentinel" by Arthur C. Clarke

Not the Future

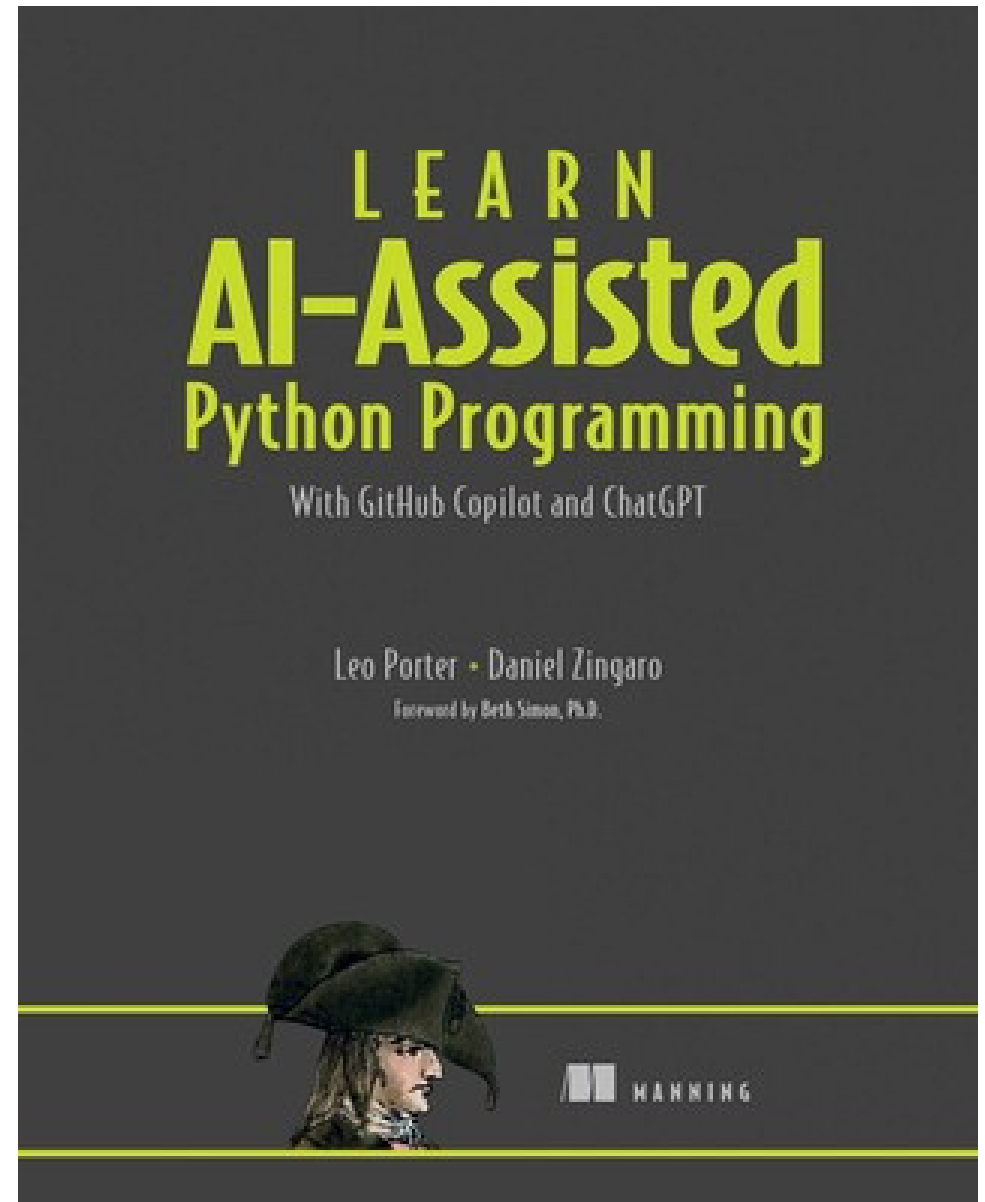
—they pictured the men of the future as having huge heads to contain very big brains. It was expected, in other words, that the future evolution of mankind would be an evolution of the mind and of the brain, and so bigger brains. But what has happened instead of that is that, instead of evolving bigness of brain, we are evolving an electronic network in which our brains are very swiftly being plugged into computer systems.

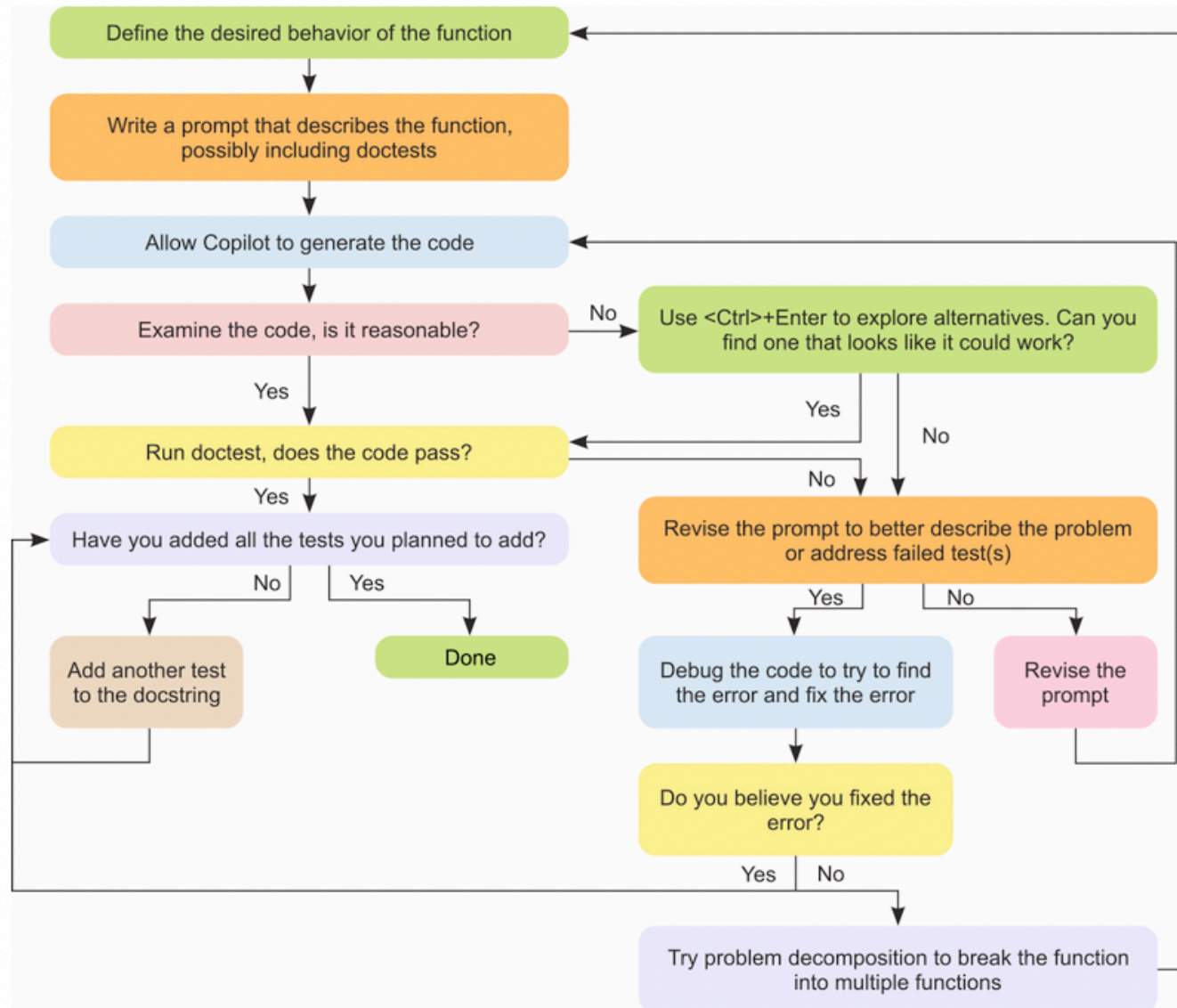
Alan Watts - <https://alanwatts.org/transcripts/future-of-communications-part-1/?highlight=electronic>



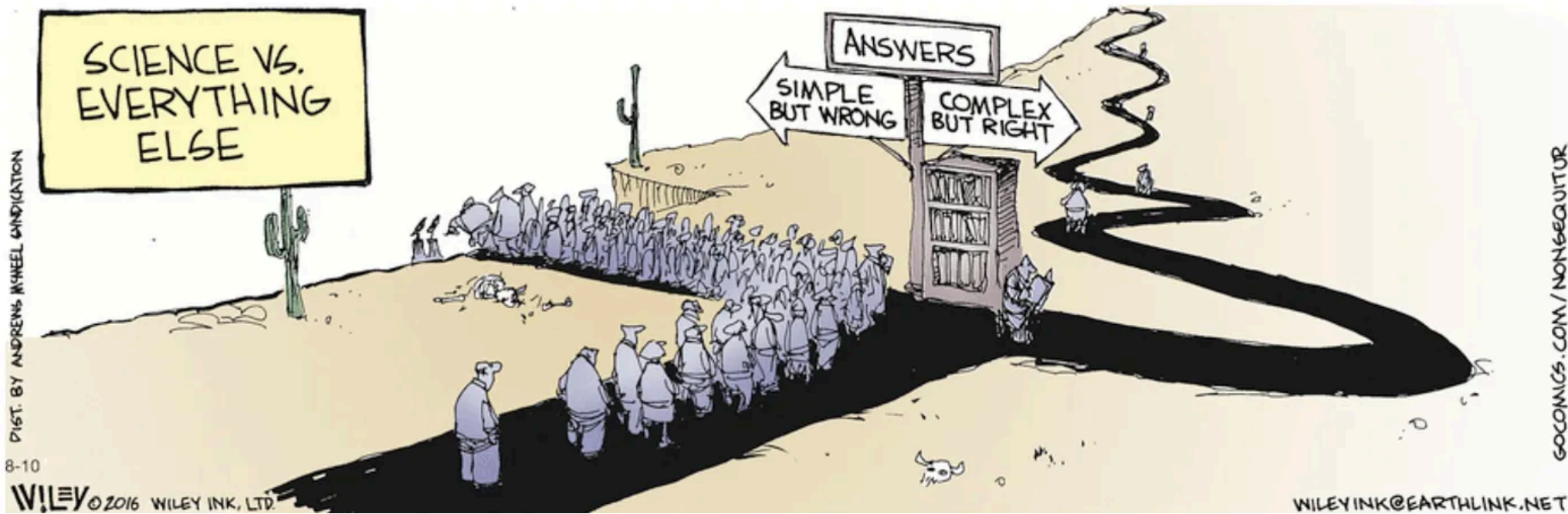
Coding with Co-Pilot

Porter, L. (2024). *Learn AI-assisted Python Programming: With GitHub Copilot and ChatGPT*. Simon and Schuster.





The function design cycle with Copilot, augmented to include debugging



DIST. BY ANDREWS McNEEL CANDIDATION

8-10

WILEY © 2016 WILEY INK, LTD.

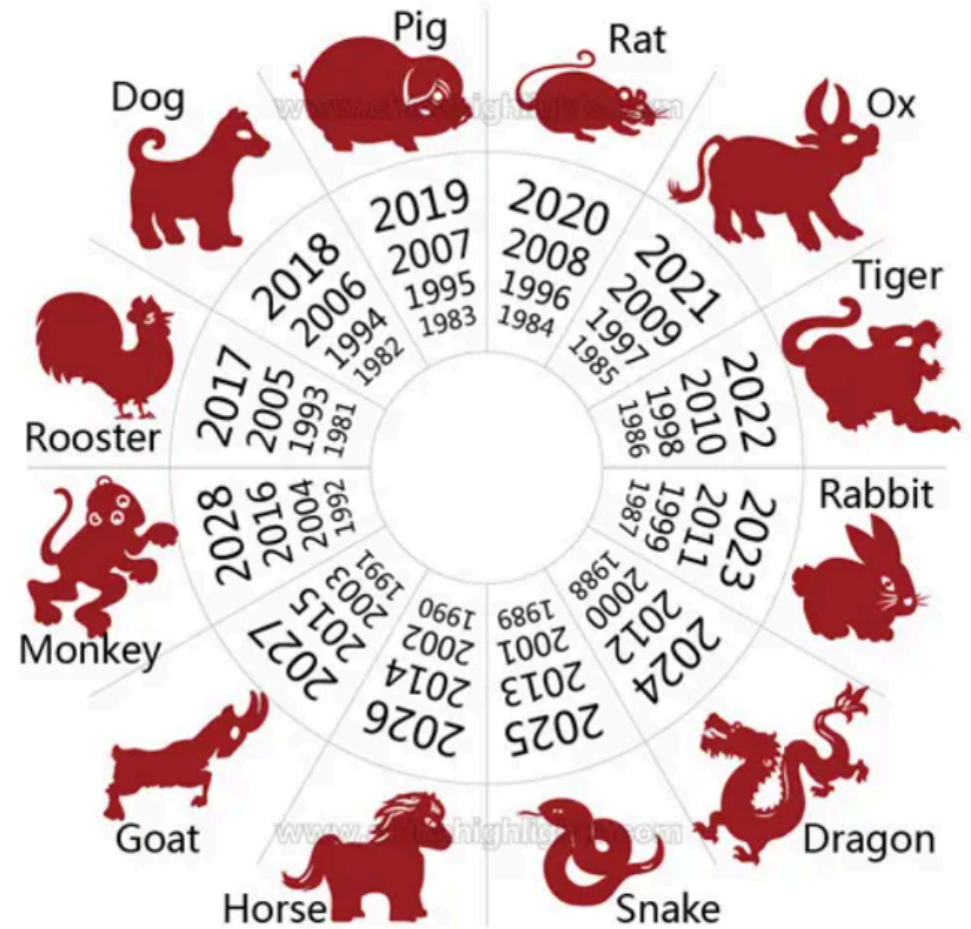
GOCOMICS.COM/NONSEQUITUR

WILEYINK@EARTHLINK.NET

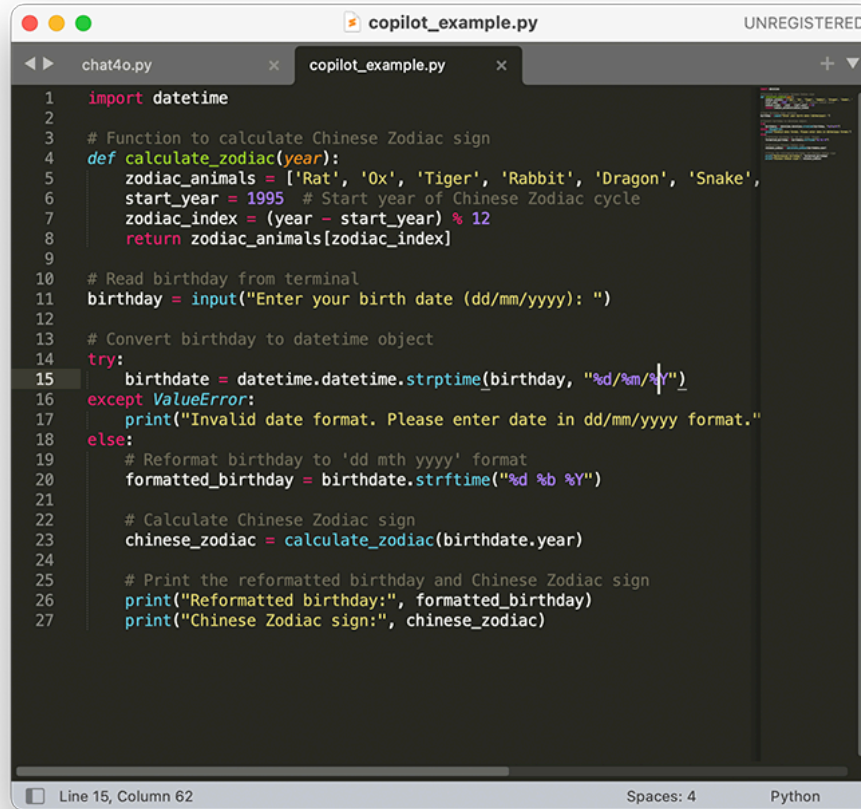
In-Class Activity Example

1. Create a Python script that calculates your Chinese Zodiac year when you enter your birth date.
2. Chinese zodiac years are represented by 12 animals. Each Chinese lunar year in the repeating zodiac cycle of 12 years is represented by a zodiac animal.
3. The order of animals in a Zodiac cycle: Rat, Ox, Tiger, Rabbit, Dragon, Snake, Horse, Goat, Monkey, Rooster, Dog, Pig.
4. Input the birthday from the terminal with format 'dd/mm/yyyy', convert it to format 'dd mth yyyy', and calculate the Chinese zodiac sign

Zodiac Table

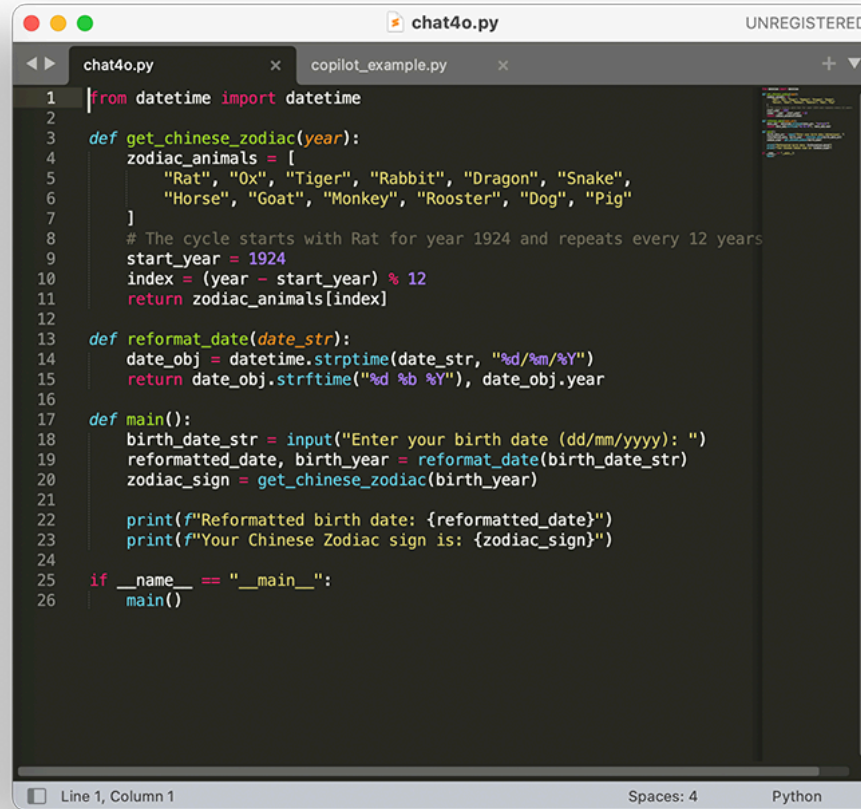


Non-Deterministic Co-Piloting...



```
1 import datetime
2
3 # Function to calculate Chinese Zodiac sign
4 def calculate_zodiac(year):
5     zodiac_animals = ['Rat', 'Ox', 'Tiger', 'Rabbit', 'Dragon', 'Snake',
6                       'Horse', 'Goat', 'Monkey', 'Rooster', 'Dog', 'Pig']
7     start_year = 1995 # Start year of Chinese Zodiac cycle
8     zodiac_index = (year - start_year) % 12
9     return zodiac_animals[zodiac_index]
10
11 # Read birthday from terminal
12 birthday = input("Enter your birth date (dd/mm/yyyy): ")
13
14 # Convert birthday to datetime object
15 try:
16     birthdate = datetime.datetime.strptime(birthday, "%d/%m/%Y")
17 except ValueError:
18     print("Invalid date format. Please enter date in dd/mm/yyyy format.")
19 else:
20     # Reformat birthday to 'dd mth yyyy' format
21     formatted_birthday = birthdate.strftime("%d %b %Y")
22
23     # Calculate Chinese Zodiac sign
24     chinese_zodiac = calculate_zodiac(birthdate.year)
25
26     # Print the reformatted birthday and Chinese Zodiac sign
27     print("Reformatted birthday:", formatted_birthday)
28     print("Chinese Zodiac sign:", chinese_zodiac)
```

Line 15, Column 62 Spaces: 4 Python



```
1 from datetime import datetime
2
3 def get_chinese_zodiac(year):
4     zodiac_animals = [
5         "Rat", "Ox", "Tiger", "Rabbit", "Dragon", "Snake",
6         "Horse", "Goat", "Monkey", "Rooster", "Dog", "Pig"
7     ]
8     # The cycle starts with Rat for year 1924 and repeats every 12 years
9     start_year = 1924
10    index = (year - start_year) % 12
11    return zodiac_animals[index]
12
13 def reformat_date(date_str):
14    date_obj = datetime.strptime(date_str, "%d/%m/%Y")
15    return date_obj.strftime("%d %b %Y"), date_obj.year
16
17 def main():
18    birth_date_str = input("Enter your birth date (dd/mm/yyyy): ")
19    reformatted_date, birth_year = reformat_date(birth_date_str)
20    zodiac_sign = get_chinese_zodiac(birth_year)
21
22    print(f"Reformatted birth date: {reformatted_date}")
23    print(f"Your Chinese Zodiac sign is: {zodiac_sign}")
24
25 if __name__ == "__main__":
26    main()
```

Line 1, Column 1 Spaces: 4 Python

Deterministic Assignment and Code-Grading

The screenshot shows a web browser displaying an assignment page. The page title is "Part 1: Hero vs. Monster Dice Game, Functions". The content includes a "Background" section, an "Objective", "Game Rules", "Requirements", "Example Output", "Tasks", and "Submission" sections. The "Requirements" section lists three tasks: 1. Dice Rolling Function, 2. Gameplay Function, and 3. Result Display. The "Example Output" shows a text box with the text: "Monster Levels: 9, 4, 3", "Hero Levels: 10, 2, 7", "Hero wins".

The screenshot shows the CodeGrade interface for the assignment. The page title is "Handin3 Due at 2023-12-11 23:59". The interface includes a "Setup" section, a "Tests" section, and a "Snapshots" section. The "Tests" section shows two test cases: "Part 1: Functions" and "Part 2: Project". Each test case has a "Pytest" section with a "Install version" of 7.3.1 and a "Code" section. The "Code" section for "Part 1: Functions" contains the following code:

```
1 # Write your bash script here.
2 mv $UPLOADED_FILES/* .
```

The "Code" section for "Part 2: Project" contains the following code:

```
1 # new test: dice 13
2 import pytest
3 import inspect
4 import hands3_2
5
6 def setup_module(module):
```

The interface also includes a "Rubric" section, a "Plagiarism" section, and a "Grades" section. The "Grades" section shows a "Setup" button and a "Tests" button. The "Plagiarism" section shows a "Snapshots" button. The "Grades" section shows a "Setup" button and a "Tests" button. The "Plagiarism" section shows a "Snapshots" button. The "Grades" section shows a "Setup" button and a "Tests" button. The "Plagiarism" section shows a "Snapshots" button.



Prediction is Easy, Judgement is Hard

Ajay Agrawal, Joshua S. Gans, Avi Goldfarb, 2019



"A science for design is by necessity both a science of making and a philosophy of realizing artifacts with and for others."

Krippendorff, K. (2006). The semantic turn: a new foundation for design (p. 349). CRC Press.5

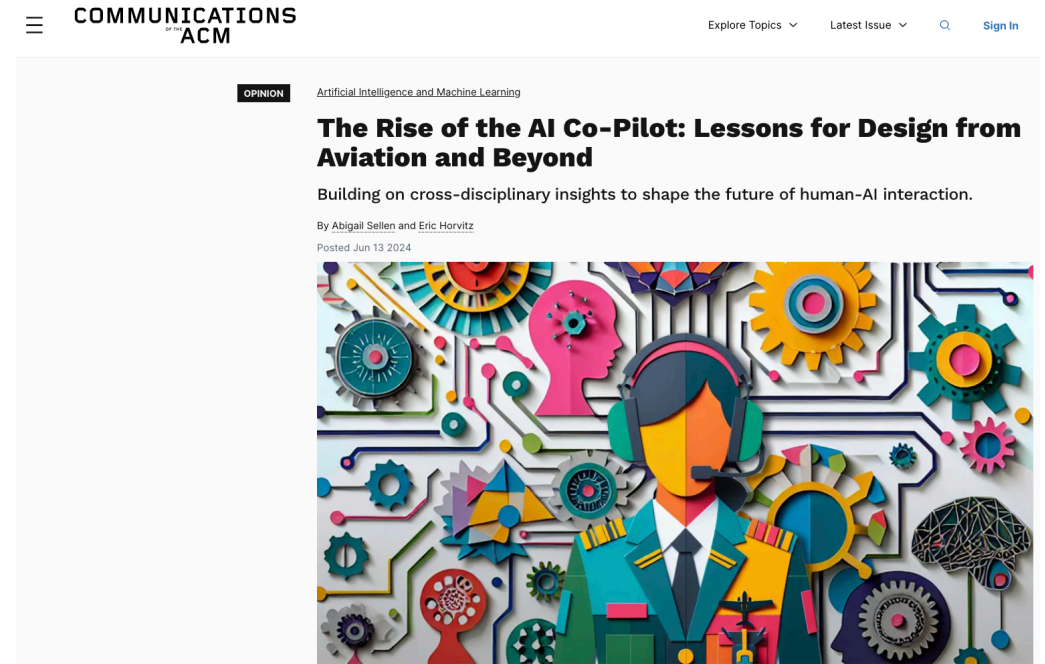
Fundamental Challenges

- Re-thinking Programming as "Design" activity
- Teaching Judgement - not syntax
- New Tools evolve faster than education

The Lessons

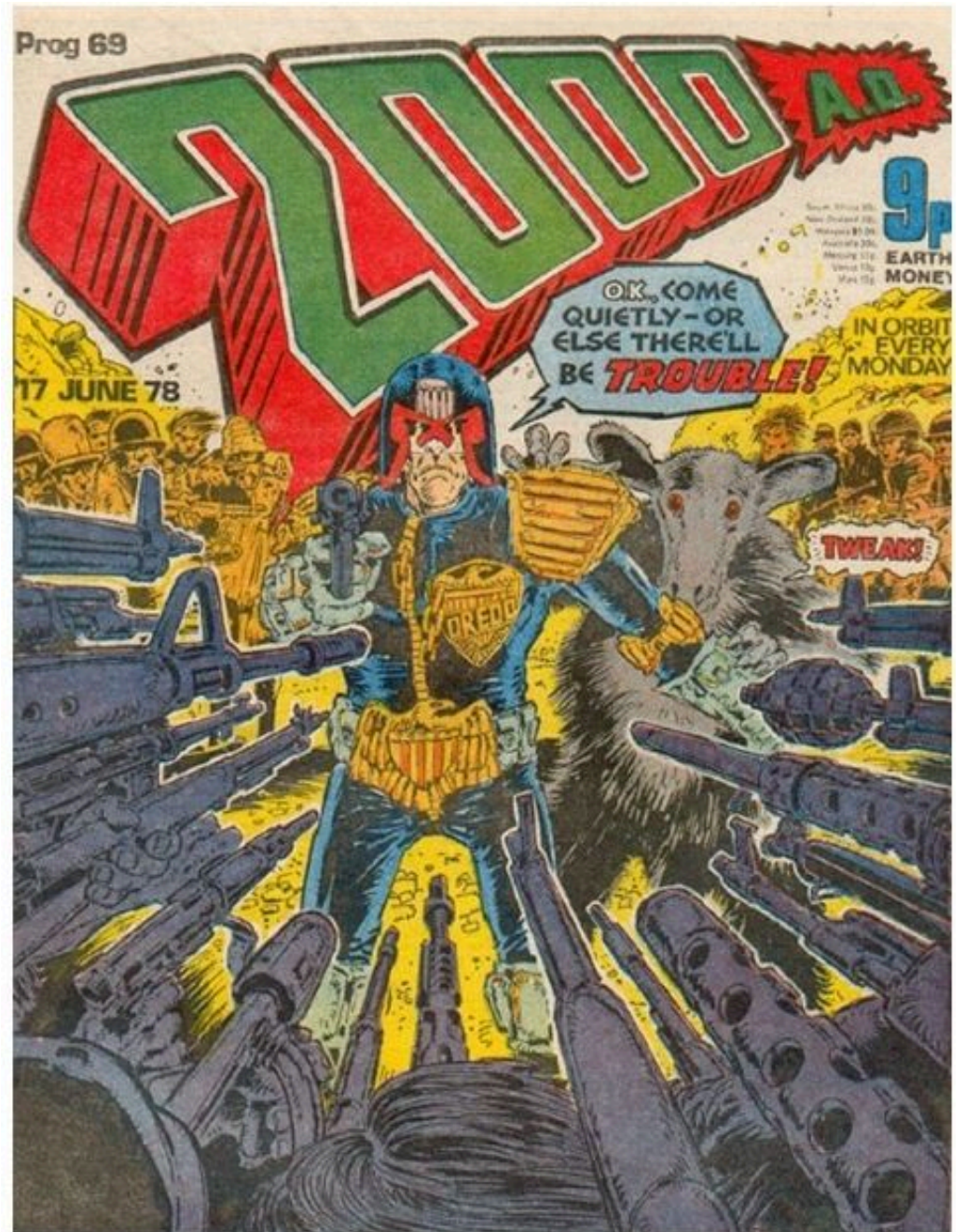
- Keep humans in the loop
- Uphold the primacy of human agency and role allocation
- Building human skills and capabilities
- Designing more intelligible systems
- Designing for appropriate levels of trust

<https://cacm.acm.org/opinion/the-rise-of-the-ai-co-pilot-lessons-for-design-from-aviation-and-beyond/>



Next steps

- Explicit connections
- Meaningful Problems
- Problem-based / Project-based
- Skills to make judgment



Corporate Intelligence

Imagine super-intelligent machines capable of transcending human abilities and superseding us. It is here - the modern corporation: AI is not a robot, but a tool primarily used for profit - Corporations mostly use humans as their sensors and effectors; they also employ us to do their work. Crucially, they lack empathy or loyalty

“Ways of Being: Beyond Human Intelligence” - James Bridle 2020

John Dewey

“Every great advance in science has issued from a new audacity of imagination...Give the pupils something to do, not something to learn; and the doing is of such a nature as to demand thinking; learning naturally results.”

John Dewey, *The Quest for Certainty: A Study of the Relation of Knowledge and Action*
1929

Background

- Saari, M., Rantanen, P., Nurminen, M., Kilamo, T., Systä, K., & Abrahamsson, P. (2024). Toward Guiding Students: Exploring Effective Approaches for Utilizing. Generative AI for Effective Software Development. doi:10.1007/978-3-031-55642-5_16
- PuryearBen, & SprintGina. (2022-11-01). Github copilot in the classroom. Journal of Computing Sciences in Colleges. doi:10.5555/3575618.3575622
- Prather, J., Denny, P., Leinonen, J., Becker, B. A., Albluwi, I., Craig, M., . . . Savelka, J. (2023). The Robots Are Here: Navigating the Generative AI Revolution in Computing Education, New York, NY, USA.
- Perry, N., Srivastava, M., Kumar, D., & Boneh, D. (2023). Do Users Write More Insecure Code with AI Assistants?, New York, NY, USA.
- Liu, R., Zenke, C., Liu, C., Holmes, A., Thornton, P., & Malan, D. J. (2024). Teaching CS50 with AI: Leveraging Generative Artificial Intelligence in Computer Science Education, New York, NY, USA.
- Liffiton, M., Sheese, B. E., Savelka, J., & Denny, P. (2023). CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes, New York, NY, USA.
- Leidner, J. L., & Reiche, M. (2024). Language-Model Assisted Learning How to Program? Communications in Computer and Information Science. doi:10.1007/978-3-031-50485-3_41
- Kazemitabaar, M., Hou, X., Henley, A., Ericson, B. J., Weintrop, D., & Grossman, T. (2023). How Novices Use LLM-based Code Generators to Solve CS1 Coding Tasks in a Self-Paced Learning Environment, New York, NY, USA.
- Jošt, G., Taneski, V., & Karakatič, S. (2024). The Impact of Large Language Models on Programming Education and Student Learning Outcomes. Applied sciences, 14(10), 4115. doi:10.3390/app14104115
- Johnson, M. (2024). Generative AI and CS Education. Communications of the ACM, 67(4), 23-24. doi:10.1145/3632523
- Hou, I., Mettill, S., Man, O., Li, Z., Zastudil, C., & MacNeil, S. (2024). The Effects of Generative AI on Computing Students' Help-Seeking Preferences, New York, NY, USA.
- Hartley, K., Hayak, M., & Ko, U. H. (2024). Artificial Intelligence Supporting Independent Student Learning: An Evaluative Case Study of ChatGPT and Learning to Code. Education sciences, 14(2), 120. doi:10.3390/educsci14020120
- Gumina, S., Dalton, T., & Gerdes, J. (2023). Teaching IT Software Fundamentals: Strategies and Techniques for Inclusion of Large Language Models: Strategies and Techniques for Inclusion of Large Language Models, New York, NY, USA.
- Damiano, A. D., Lauría, E. J. M., Sarmiento, C., & Zhao, N. (2024). Early Perceptions of Teaching and Learning Using Generative AI in Higher Education. Journal of educational technology systems, 52(3), 346-375. doi:10.1177/00472395241233290
- Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023). Programming Is Hard - Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation, New York, NY, USA.