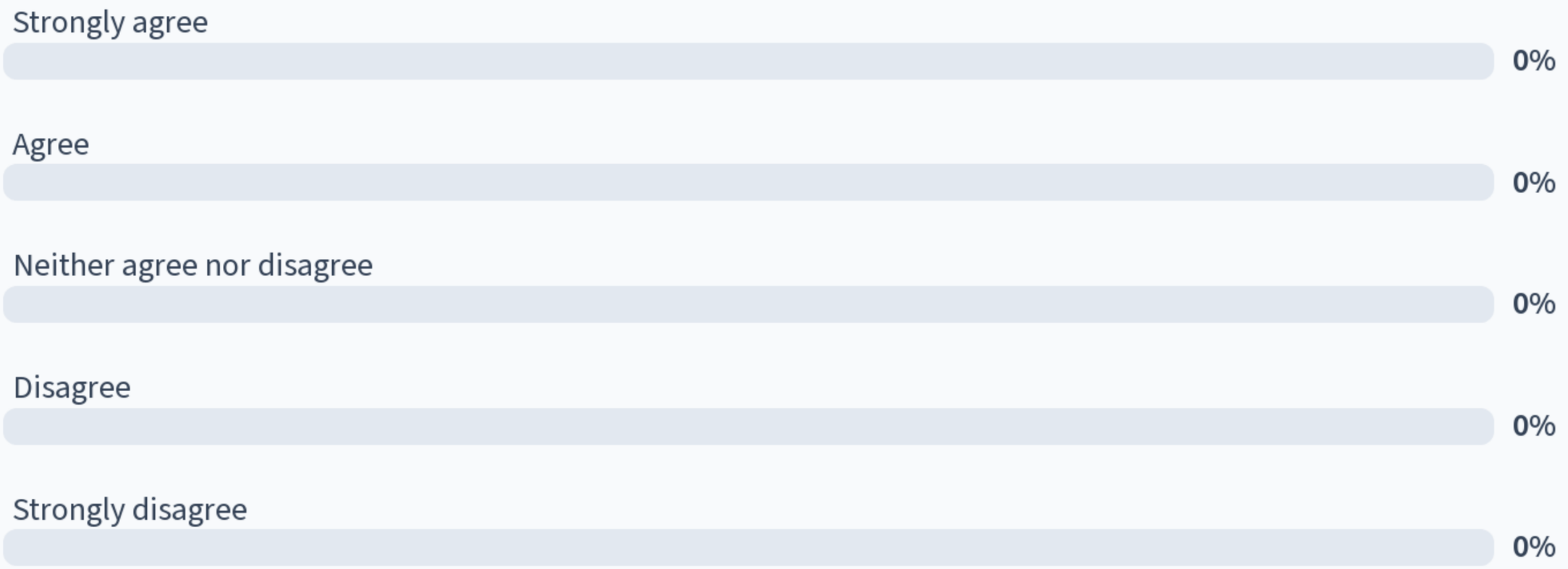
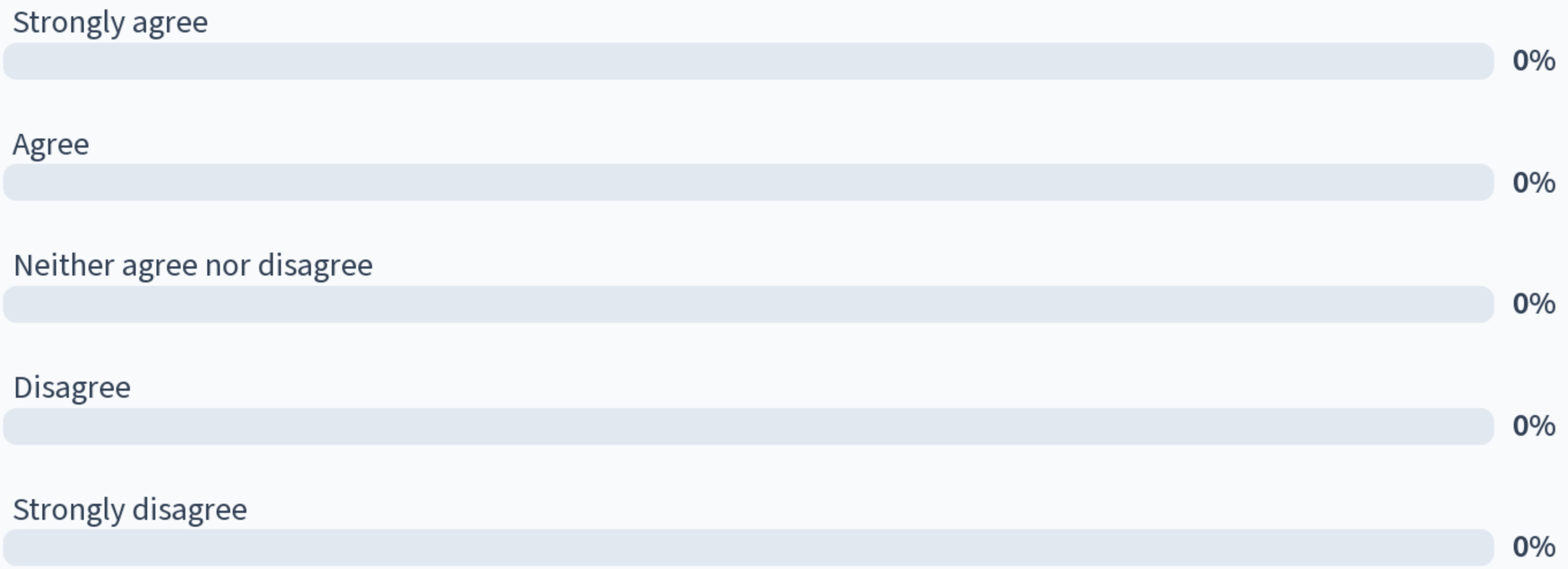
Intellectual Property for Software Engineers

All software should be free to use



A company should be able to profit by selling open source software



Today

- Software engineering requires understanding basic principles of intellectual property
  - You will likely need to select and interpret software licenses
- Disclaimer: I am not a lawyer!
- Info today may be US-centric

Basic Terms

- **Trademark**: protecting terms from misuse
- **Patents**: protecting inventions (ideas)
- **Copyright**: protecting work from copying
  - Software licensing: restricting what you can do with a copy

Trademark (Source: US PTO)

- "A trademark can be any word, phrase, symbol, design, or a combination of these things that identifies your goods or services. It's how customers recognize you in the marketplace and distinguish you from your competitors."
- "A trademark:
  - Identifies the source of your goods or services.
  - Provides legal protection for your brand.
  - Helps you guard against counterfeiting and fraud."

Trademark Examples

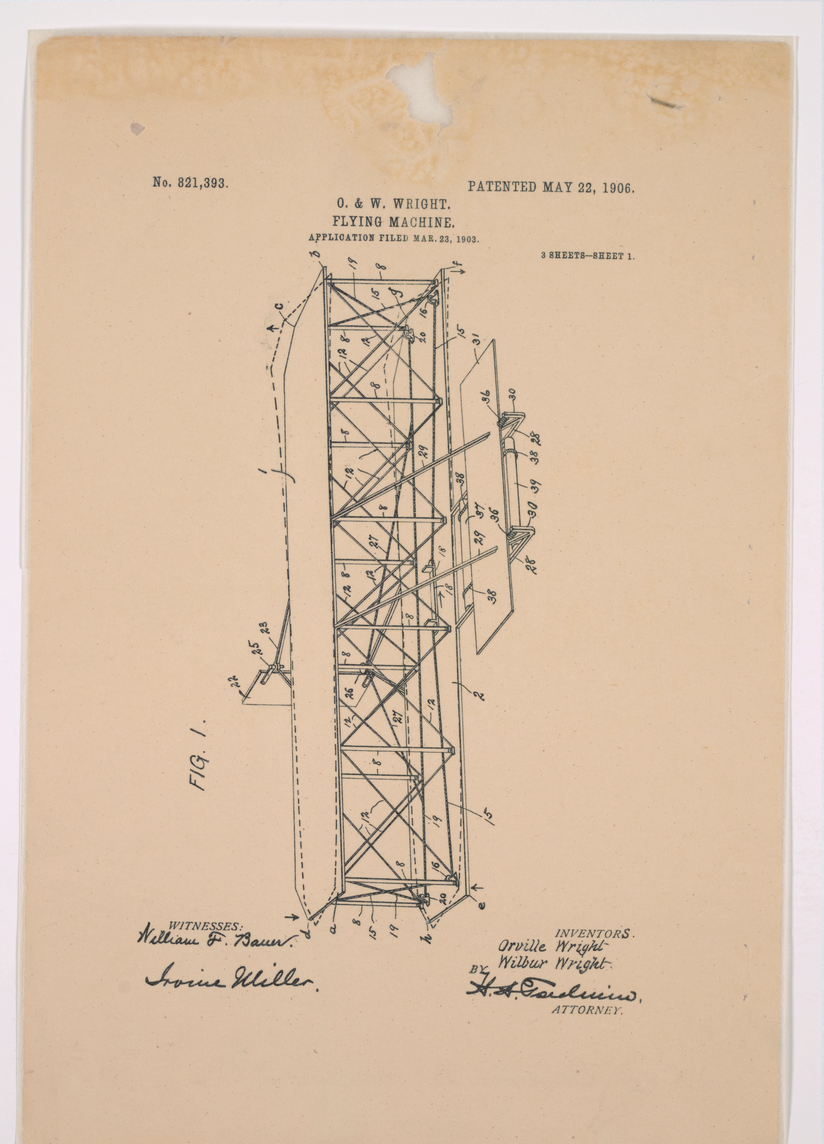
- "Domino's" is a trademark of a pizza company
- You can make and sell children's toys, "dominoes," and not get sued
  - Trademarks are connected to specific goods or services
  - Domino's pizza trademark pertains to food, not toys

Patents

- Patents cover inventions
- Today: focus on *utility* patents (vs. design, plant patents)
  - "These may be granted to anyone who invents or discovers a new and useful process, machine, article of manufacture, or composition of matter, or any new and useful improvements of these." (USPTO)
  - Patents are supposed to cover "non-obvious" inventions

Patents

- At right: diagram from patent for Wright brothers' flying machine (1906)
- Principle: promote invention by allowing inventors to profit from their inventions
  - Others making a flying machine with the same design would invite a lawsuit by the Wright brothers (patent holder is responsible for enforcement)
- Patents *expire* (usually after 20 years)
  - Allows others to leverage invention for free
  - Example: generic drugs (cheaper than brand name)


No. 821,393. PATENTED MAY 22, 1906.

O. & W. WRIGHT.

FLYING MACHINE.

APPLICATION FILED MAR. 23, 1903.

3 SHEETS—SHEET 1.

FIG. 1.

WITNESSES:
William F. Bauer.
Irvine Miller.

INVENTORS.
Orville Wright.
Wilbur Wright.
BY
ATTORNEY.

Software Patents

- Controversial
- Can't patent math (e.g. algorithms). Can patent machines. Where is the boundary?
- My system is composed of a zillion components. How do I tell whether any of them are patented?
- Does patenting encourage or discourage innovation?

Software Patent History

- Gottschalk v. Benson, 1972: allowing patents on software might actually patent an algorithm, but math is abstract and therefore not patentable
- 1981: Diamond v. Diehr: An invention that included software was ruled patentable (but invention included steps related to heating rubber): this was a *process* for molding rubber
- Question: when is an invention "merely" mathematical?

https://www.bitlaw.com/software-patent/history.html

- 1998: Federal Circuit court upheld a patent about business methods (relating to running mutual funds): software that yields a useful, concrete, and tangible result are patentable

- In re Bilski, 2008: a process is patentable if "(1) it is tied to a particular machine or apparatus, or (2) it transforms a particular article into a different state or thing."

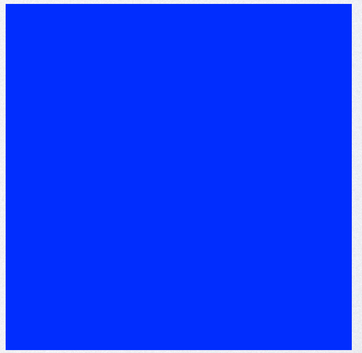
- Supreme Court, 2010s: actually, the 2008 test is only a clue, not a test! (collection of different cases)

Copyright

- Protects **original works of authorship** as soon as an author **fixes the work** in a **tangible form of expression.**

- "The Supreme Court has said that, to be creative, a work must have a "spark" and "modicum" of creativity."

- Can't copyright this blue square:



Source: https://www.copyright.gov/what-is-copyright/

Fixing

- If I sing a new song, but no one records it, it is **not copyrighted**
- …but it is copyrighted as soon as it is recorded!
- These lecture slides are copyright 2024, Michael Coblenz
  - Though usually an employer owns an employee's works

Reference: https://commons.ucsd.edu/_files/remote_instruction_copyrights

Rights of Copyright Holders

- Reproduce the work in copies.
- Prepare derivative works based upon the work.
- Distribute copies of the work to the public by sale or other transfer of ownership or by rental, lease, or lending.
- Perform the work publicly if it is a literary, musical, dramatic, or choreographic work; a pantomime; or a motion picture or other audiovisual work.
- Display the work publicly if it is a literary, musical, dramatic, or choreographic work; a pantomime; or a pictorial, graphic, or sculptural work. This right also applies to the individual images of a motion picture or other audiovisual work.
- Perform the work publicly by means of a digital audio transmission if the work is a sound recording.

Source: https://www.copyright.gov/what-is-copyright/

Copyright vs. Patents

- Key contrast with patents:
  - If I independently replicate your invention, I infringe a patent anyway
  - If I independently reproduce your copyrighted work, no problem; I didn't copy

Licensing

- What if you want to allow others to use your copyrighted works?
- I want you to be able to execute my code
  - Optionally: you have to pay me first
  - Optionally: but you can't share the code with anyone else
  - Optionally: also, you can create derivative works

Public Domain

- Copyright lasts up to 120 years from creation (works made for hire)
  - Or 70 years after your death (works not made for hire)

Which of the Following Scenarios Is Most Likely Eligible for Software Patent Protection in Jurisdictions That Allow Software Patents?

A. A software developer creates an algorithm for sorting numbers, similar to existing methods but implemented in a different programming language.

B. A software company designs a novel method for encrypting data that achieves faster processing times and greater security than existing techniques.

C. An app developer writes original source code for a social media application with standard features like messaging and user profiles.

D. A software engineer writes an operating system optimized for specific hardware..

A Case for Open Source

- Many eyes make bugs easier to find
- End users can improve and customize software
- Code reuse improves productivity

A Case Against Open Source

- Who will create software if no one will pay for it?
- Open source code may be easier to attack
- Innovation may be stifled if software can just be copied

Copyleft

- What constitutes free software?
- Key question: can I make proprietary changes to your free software?
- Copyleft says: no, all derivative works must remain copyable and changeable
- Alternative: derivative works may have additional restrictions
- Generally: copyleft software can't be used in proprietary products
  - since otherwise the company would have to make their code open source

Free Software Philosophy ("Free as in Speech, Not As in Beer")

- Freedom 0: The freedom to run the program as you wish, for any purpose.
- Freedom 1: The freedom to study how the program works, and change it so it does your computing as you wish.
- Freedom 2: The freedom to redistribute copies so you can help your neighbor.
- Freedom 3: The freedom to distribute copies of your modified versions to others

Activity

- Form groups of 3
- Read:
  - BSD, 3-clause
  - GPL v3
  - MIT

- For each:
  - Proprietary use?
  - Modify?
  - Give away patent rights?
  - Must include copyright notice?
  - Share source of derivative works?
  - Warranty provided?