Today

- Quiz
- Legal Issues: Intellectual Property
  - Software Copyrights (part 1)
  - Software Patents (part 2)

Homework

- Exercise Set 7 due
- Exercise Set 8 out (due at Final)
- Essay’s 1–3 out (due at Final)
Exercise 1: What should your rights be after purchasing a software product?

Consider:

- using the software
- copying the software for use by yourself
- copying the software for use by others
- modifying the software for your own personal use
- reselling modified software
Copyright Law

Protecting “creative works” from unauthorized copying

Copyright protection is for “original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device.”

• in general protects the author’s “expression”

• not the idea behind the expression (patents)

Copyrights give the copyright holder exclusive rights (*) to:

• Make copies of the work
• Produce derivative works (e.g., translations, movies/books)
• Distribute copies
• Perform the work in public
• Display the work in public

(*) With some exceptions based on “fair use” rules ...
The Fair Use Doctrine

Copyright law has two main goals:

1. Promote production of useful work (creator protections)
2. Encourage the use and flow of information (use protections)

Fair Use provides rights to users of copyrighted works ... e.g.:

- criticism, commentary, news reporting, teaching (excerpts)
- scholarship and research
- basic use as a consumer

Four broad factors when considering fair use cases ... 

1. Purpose and nature of the use
   - commercial less fair use than nonprofit or educational
2. Nature of the copyrighted work
   - creative works are less fair use than factual
3. Amount and significance of portion used
4. Effect of the use on potential market or value of work
   - uses that reduce sales are less fair use
Copyright Law

Copyrights last for a limited time period:

- The lifetime of the owner plus 70 years (*)
- After this, the work becomes part of the public domain
- Where anyone can freely copy and use the work

(*) Unless created from 1923–1978, then 95 years

Copyright Law and Software: Some History ...

- Software not originally considered protected
  - not seen as “fixed, tangible objects”
- In 1976, computer programs included (under Copyright Act)
- Courts later clarified programs as “literary works”
- Various court cases in 80’s and 90’s
- Digital Millennium Copyright Act (DMCA) in 1998
Copyright Violation

Copyright holders can seek redress (sue for damages)

- any member in the chain of distribution
- regardless if they knowingly/unknowingly violated copyright
- with some limits (fair use, and DMCA “safe havens”)

Proving copyright infringement requires proving:

- Ownership of the copyright
- And that copying took place

To prove copying took place can show directly, or else show:

- The violator had access to the copyright material
- Substantial similarity between original and copy

Exercise 2: With a partner, develop rules for defining:

“Substantial similarity between original and copy” for software
Copyright Violation

Courts had to develop ways to determine software violations

- The primary function was treated as the main idea (couldn’t be copyrighted)
- Everything else not necessary for the primary function was expression
- Evolved into the AFC test (abstraction-filtration-comparison)

The Abstraction-Filtration-Comparison (AFC) test ...

1. Identify increasing levels of abstraction
   - Lowest level is code (copyrightable)
   - Middle levels (groups of instructions, modules, etc.)
   - Highest level is main function (not copyrightable)

2. Remove aspects not covered by copyright (filtration)
   - E.g., elements for efficiency, elements dictated by external factors (standard techniques/designs), elements taken from public domain

3. Compare each element under copyright (comparison) for violation
Rights of Consumers / Users

Fair Use allows, without authorization of copyright holder ...

“Owners” of copies to make additional copies for

- Archival or maintenance purposes
- Utilization of a computer program (a step in using)

“Owners” of copies to resell their copy ("first sale" doctrine)

- The rights are then exhausted by the old owner
- “Owners” of copies to rent their copy
- Again, rights are exhausted by owner until returned

“Owners” of copies to modify the copy for personal use

“Owners” of copies to reverse engineer their copies

Among others ...

(*) Note that most software is now licensed not purchased

- Meaning as a purchaser, you don’t own the copy
- Courts have ruled differently on purchased vs licensed
Software Copyright Court Cases

Sony vs Universal Studios (1984)

- Recording movies on TV using Sony Betamax machines
- Sued Sony because machines used to do the copying
- Sony won ...
  - Copying largely for private, non-commercial use
  - Even though entire (not a portion of) movies copied
  - Makers of a device that some use to infringe copyright, but with legitimate uses should not be penalized

Sega vs Accolade (1992)

- Sega had large licensing & costs for 3rd party developers
- Accolade made video games for the Sega console
- And bypassed costs by reverse engineering console software (copied and decompiled)
- Sega sued, Accolade won:
  - Court viewed purpose as creating new works
  - Court viewed this as fair competition, even though it could reduce Sega’s game market (not console market)
Napster vs Recording Industry Assoc. of America (2001)

- Issues:
  - Is P2P copy & distribution of music legal under fair use?
  - If not, is Napster responsible for the actions of its users?

- Napster argued copies for personal, not commercial use
  - Personal implies limited (not distribution to the masses)
  - RIAA argued large impact to music industry/market

- Napster lost on both issues

Apple vs Microsoft (1988)

- Apple sued Microsoft to prevent them from using their GUI elements

- Apple lost
  - Apple sued at the same time by Xerox who also lost

- “Look and feel” cases have largely been struck down
  - Limited protections under design patents do exist
Digital Millennium Copyright Act (DMCA)

Digital rights management (DRM)

- Techniques to control access to and uses of digital IP
- Includes hardware and software using encryption
- Embedded into text files, music, movies, ebooks, etc.
- Prevents saving, printing, copying, creating excerpts, etc.

DRM tightly controls use ... including in some cases fair use

Anticircumvention provisions

- Prohibit making, distributing, using tools to circumvent DRM
- Goal is to reduce piracy and other illegal uses of IP
- Problem is it criminalizes actions that don't infringe copyright

Various legal cases, some exceptions (see book)

- E.g., for security research
- For some devices (e.g., phones, CDs)
Safe Harbors

- Protects providers from copyright violations of users
- Including ISPs and web sites (e.g., YouTube)
- Cannot profit from the material
- Must remove material when asked (take down notice)
  - Has led to various issues ...
  - E.g., YouTube receives millions of requests per year
US Patents

What is a patent?

Ability to legally claim rights to an “invention” or “discovery”

- Essentially give patent owner a legal monopoly

... the right to exclude others from making, using, offering for sale, or selling the invention or discovery in the United States or “importing” the invention or discovery into the United States

For example ...

- If you invent something that has been patented,
- you cannot make, use, or sell the invention without consent of patent owner
- typically by paying royalties/licensing fees

The purpose of allowing patents ...

- To encourage innovation (by protecting inventions)
- But unlike copyright, doesn’t give rights to consumers
More on US Patents

What is a patent? (cont)

• Ability to legally claim rights to an “invention” or “discovery”
• Granted through US Patent & Trademark Office (USPTO)
• Typically last 20 years from filing date
• Patent disputes handled in courts, not by USPTO
• Effective within the US (other countries have similar)

⇒ Important:

• USPTO grants patents ...
• but courts determine whether patents are valid (via lawsuits)!
What can be patented?

There are 3 types of patents:

1. **Utility patents**: process, machine, article of manufacture, or composition of matter, or an improvement thereof
   - Most software patents are utility patents

2. **Design patents**: design for a functional (utilitarian) item. The ornamental appearance, not structure or features. Copyrights are for non-utilitarian items.
   - Only last 14 years.
   - Examples: Coke Bottle, iPhone, Statue of Liberty, fonts
   \[ \Rightarrow \text{Utilitarian implies useful or practical} \]

3. **Plant patents**: plants as in crops, bushes, trees, etc
Patent Eligibility Constraints

Invention must be **new** (novel/original) and **useful** (specific purpose)

1. Not on the abstract idea, but on new machine, manufacture, process, etc.
   - An abstract idea is not patentable ...
   - Neither are laws of nature or mathematical formulas

2. A complete description of the machine or subject matter is required
   - typically through a set of “claims”

3. Not considered new if already claimed or available to the public before filing
   - e.g., publication, presentation, sale, etc.

4. New also means **non-obvious**
   - Often contentious for software patents (and others)
Software Patents

How do patents apply to software?

- Can file for both utility and design patents
- Utility patents much more common (and easier to uphold)
- Google, e.g., patented the design of their webpage

Exercise 3: With a partner consider utility patents for software / CS ...

- **Utility patents**: process, machine, article of manufacture, or composition of matter, or an improvement thereof
- How would this apply to software / CS?
- What types of claims might be made?
- How would new/novel & useful be judged?

The state of software patents today ...

- Hard to distinguish patentable vs non-patentable
- Non-obvious criteria has been loosely interpreted
- Abstract idea vs process/machine loosely interpreted (*)
- Issues in encouraging innovation (e.g., “patent trolls”)
- USPTO overwhelmed with number of software patents

(*) This has changed recently in courts
Issues in software patents

Non-obvious criteria has been loosely interpreted

• Has led many notoriously bad patents!

• See, e.g., https://www.eff.org/issues/stupid-patent-month

Patents are often used as “offensive & defensive weapons” ...

• Google, Apple, Microsoft paid billions of dollars for patents related to smartphones (100,000s of patents)

• Purchased to defensively counter-sue companies that sue them for patent infringement

• As well as to offensively suppress or limit competition
Alice Corp vs. CLS Bank International (2014)

Context ...

- Alice Corp owned 4 patents for managing risk in financial trading systems
- They filed suit against CLS Bank for infringement
- CLS Bank filed suit against Alice Corp seeking judgement on whether the patent claims were valid/invalid

The case went to the Supreme Court ... the main issue:

- Was the patent really just for an abstract idea ...
- That can or happens to be implemented in a program
- Or a practical & innovative implementation of an idea

Based on previous cases, the court rule both are needed:

- An abstract idea (e.g., algorithm, computation method, process, ...)
- And a practical, innovative implementation of idea (“innovative concept”)
- Implementation can’t be generic, conventional, or obvious

The court invalidated the patents

“merely requiring generic computer implementation fails to transform an abstract idea into a patent-eligible invention.”
The state of software patents today ... (post Alice ruling)

- Decline in “business process” software patents since Alice ...
- > 50% fewer issued (per month) after Alice
- ≈ 30% jump in rejections due to Alice
- More patents invalidated ...
- e.g., ≈ 78% of patent challenge cases resulted in invalidated patents