Towards a working HARPOON

- **Get the design right**
  - what is a figure? Develop the proper abstraction
  - where is a figure drawn? What is page?
  - with what tools does the user interact with the program, what is a tool?
  - develop use cases, how does user interact with program, these should be documented and included in your final project

- **What is a figure, what is page? (CRC at best, ideas at least)**
Saving and restoring objects

- Classes should implement Serializable, this is a tag interface, not necessary to implement a function (see Cloneable)
  ➤ see Nutshell examples, Chapter 13, see ProtoHarp
  ➤ mark non-serializable fields as *transient*
    • platform specific objects like font sizes, these need to be reconstructed rather than re-read
    • fields that aren’t needed when an object is deserialized
  ➤ use ObjectOutputStream and ObjectInputStream, can customize behavior using private?! functions below
    ```java
    private void writeObject(java.io.ObjectOutputStream out) throws IOException
    
    private void readObject(java.io.ObjectInputStream in) throws IOException, ClassNotFoundException;
    
    ➤ also defaultReadObject() and defaultWriteObject()
    ```
Issues in save and restore/cut and paste

- A vector is serializable, what about a vector of figures?
  - What if Figure extends Component? Is that enough?
  - Possible to throw NotSerializableException
  - what about a vector of pages of vectors of figures?

- How is a figure saved to the clipboard?
  - See ProtoHarp and the class FigureSelection, some information accessible in Graphic Java book as well
  - note: function classForName, this is part of Java’s reflection package, possible to configure program at runtime!

- How can you select multiple figures and cut/copy/paste these?
Towards being a hacker

● **See the hacker-faq (cps 108 web page)**
  ➤ Hackers solve problems and build things, and they believe in freedom and voluntary mutual help. To be accepted as a hacker, you have to behave as though you have this kind of attitude yourself. And to behave as though you have the attitude, you have to really believe the attitude.

● **The world is full of fascinating problems**
  ➤ no one should have to solve the same problem twice
  ➤ boredom and drudgery are evil
  ➤ freedom is good
  ➤ attitude is no substitute for competence

*You may not work to get reputation, but the reputation is a real payment with consequences if you do the job well.*
Aside: ethics of software

- What is intellectual property, why is it important?
  ➤ what about FSF, GPL, copy-left, open source, ...
  ➤ what about money
  ➤ what about monopolies

- What does it mean to act ethically and responsibly?
  ➤ What is the Unix philosophy? What about protection?
    What about copying? What about stealing? What about borrowing?
  ➤ No harm, no foul? Is this a legitimate philosophy?

- The future belongs to software developers/entrepreneurs
  ➤ what can we do to ensure the world’s a good place to be?