Introduction to Computer Science and Programming in C

Session 22: November 20, 2008

Columbia University

Announcements

- Homework 4 is out, due last day of class: December 4 before class
- Final Review Thursday 12/4
- Last non-review class: free topic
- Final Exam: Tuesday, 12/16, 1:10 pm - 4:00 pm Mudd 233 (our normal room)

Review

- Data structures
  - Linked Lists: structs with pointers to next
  - Doubly linked lists: also previous pointer
  - Binary Trees: structs with pointers to left and right children.
    - Left child always less than parent, right child greater

Today

- Totally random tidbit: modulo
- A glance at Software Engineering

Modulo

- Another operator (+-*/) %
- a%b
  The remainder after dividing a by b
- 5 % 3 = 2
  5 / 3 = 1, with a remainder of 2
- Useful if you are iterating and you want something to happen every few iterations

Modulo

```
for (i=0; i<N; i++) {
    if (i % 5 == 0)
        printf(“Five iterations have passed\n”);
    <do other stuff>
}
```

Software Engineering

- What is software engineering?
- Methodologies for managing large software projects
- Think of large pieces of software as construction projects: engineering is more than programming

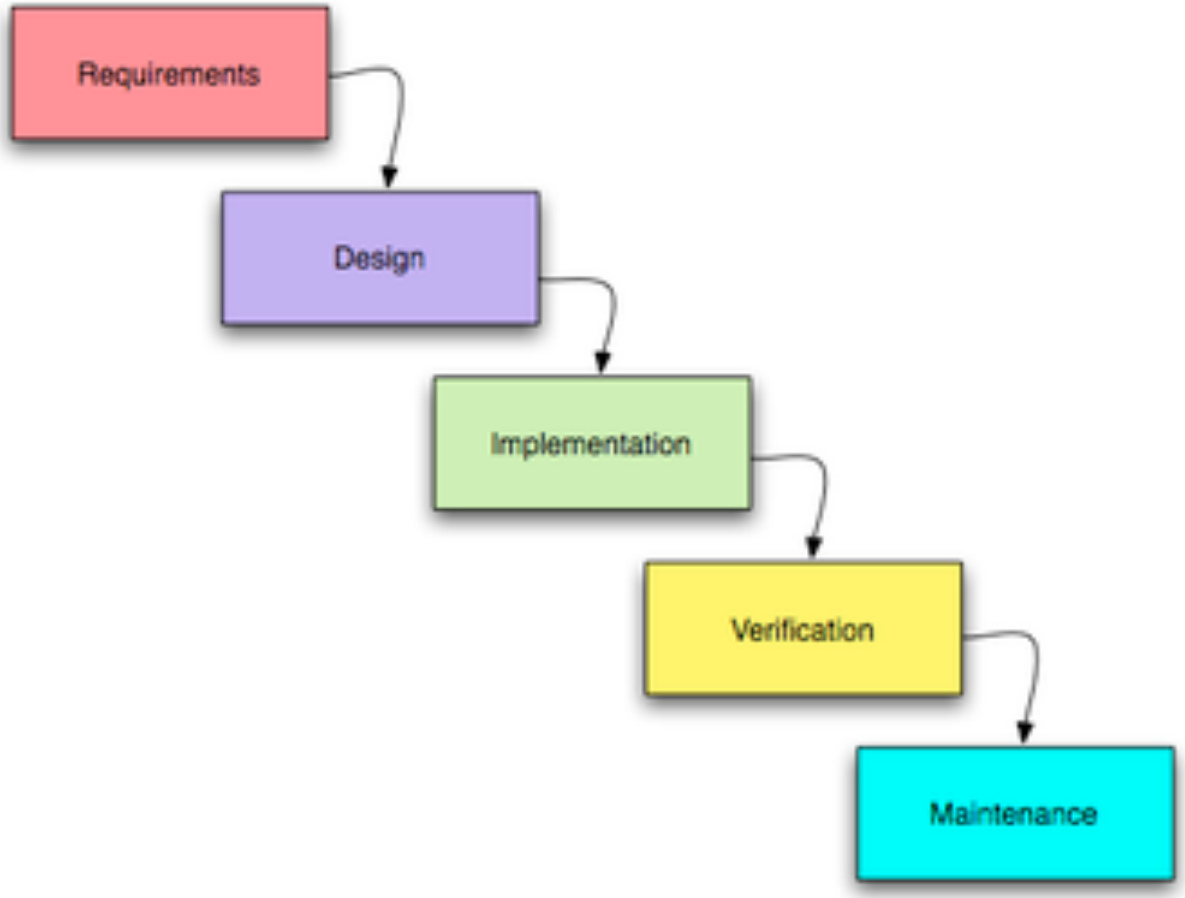
Software Engineering

- Models for software development have changed significantly since early computers
  - Programs are more complex
  - Programmers are more abundant

Basic Framework

- "Customers" and developers decide on **requirements**
- Developers write software
- Developers deliver software to customer

Waterfall Method



Waterfall Method

- Requirements: developers and customer decide what the software will do
- Design: developers plan how to program
- Implementation: developers program
- Verification: developers test program
- Maintenance: developers deliver to customer and continue to fix and update software

extreme Programming (XP)

- Popular newer methodology
- A version of Agile Programming
- Encourages more interaction between customer and developers
- Customer becomes part of the team

User Stories

- Rather than writing out long descriptive requirements, write short user stories
- Should fit on index card
- Describe a hypothetical experience of a user with the finished software

Pair Programming

- Always program in pairs
- Partners catch each others' mistakes
- Motivates programming with less distraction
- XP claims that the time saved catching each others mistakes is more than time lost by using two people

Test First

- First write software test procedures before you write the actual software
- Gives tangible target: when all tests pass, you are done!
- Think of unwritten parts of your program as parts that fail their tests

Refactor

- Refactoring means redesigning the structure of your software, moving abstractions around, etc
- Continually refactor to make code more modular.
- The tests will verify if your refactoring breaks anything, so go nuts with refactoring

Spikes

- Quick design experiments to see if an idea works
- Rather than embarking on huge piece of software, design a smaller piece and spend a few days working on that
- Find out what doesn't work quickly; practice what does work

Iterations

- Design user stories
- Implement a few user stories first
- Show result to customer
- Redesign user stories if necessary and repeat

Free Topic

- On our last non-review class, I want to give a lecture on a free topic in computer science
  - Artificial Intelligence + Machine Learning
  - Theoretical Computer Science
  - Computer Vision + Digital Signal Processing
  - Network Security
  - Computer Graphics
  - Operating Systems