

211Proposed

Proposal for New Curriculum

Task force: Walid, Corky, Alan, Joe, Dung, (Steven?)

Week	Lecture	Theme	Content	Lab
1	1-3	Functional programming + types upto trees	Booleans, records, unions, recipe, (auxiliary) functions, lists	
2	4-6		Complex list operations, natural numbers	
3	7-9		Trees, mutual recursion, GUI programming	
4	10-12		Abstraction in functions and types, first class functions	
5	13-15		Accumulators (list reverse), non-structural induction (parsing)	
6	16-18			
7	19-21	Design patterns (transition to Java)	Immutable list, composite, interpreter, helpers, visitor pattern	
8	22-24		Inner classes, GUI, MVC, generics, collections	
9	25-27			
10	28-30			
11	31-33	Programming with imperative features		
12	34-36			
13	37-39			
14	40-42			
15	43-45			

TODO:

- Can we postpone the effects of 210 to later in the course?
 - Yes, but we have to make sure we replace the GUI fun stuff by something else
 - Idea: let's provide them with a GUI harness that takes a functional program from them, and the functional program animates the GUI
- Specifying lab projects and weekly homework goals
- More explicit statements of motivation for different parts of the course
- Gain more experience with language levels (Walid)

Things for the second half:

- Iterative refinement

Falling off the cliff:

- Mutual recursion
- Currying/uncurrying
- Graphs
- Backtracking

Ideas:

- Corky: Would like to use a boolean simplifier as a driving example

Concerns:

- Walid: The functional programming sounds like it will be boring.
 - Alan: Motivate with 3 interesting problems from start?
- Joe: Feedback from students suggests we **really** need to emphasize **why** they are learning this material
 - Walid: I've improved this aspect of the course a bit last semester. We need to continue and add to this direction. It would be good to spell some of this motivation in this proposal.
- Joe: Java issue:
 - Language levels. People are split on that.
 - We need to say something like "language levels let you isolate muscles, as in weight training"
 - Need to introduce lab sessions aimed specifically at language newbies
 - Walid: It would be good to have a teachers manual for a course like this, and to develop a community of teachers.
- What should we do for the final lab?
- Dung: Spending too much time on functional programming in Java might seem contrived to students. It doesn't seem to save us any time when we starting teaching OO design.
 - Walid: The purpose of the transition is to make sure that students understand that the recipe approach evolves into the pattern-based approach in OO, and that patten-design is much easier in the functional setting. (In particular, what we want to preserve is not the understanding of lambda, but the **recipe**).
 - Functional programming is just a backdrop to help us explain data-driven design and the recipe more clearly.
 - Dung: We will need to explain how we express the datatype in Java, namely, using inheritance and composition. We really need to work on the details of how we explain this transition to students.

- Joe: Transition to Java requires caution:
 - Syntactic hurdle
 - Risk of students thinking "Rice research sucks"
 - Syntax on a "need to know basis" tends to really annoy students
- Dung: The transition will be a risk, and it will be important to do this right for the course to succeed.

References

- [COMP 210 Principles of Programming & Computing](#)
- [COMP 212 Intermediate Programming](#)
- [SIGCE 2005 Tutorial on DrJava](#)
- [ProfessorJ](#)

Reference materials

```
Corky's independent check:
Condensing the Comp 210 Syllabus

Lecture 1    Intro
Lecture 2    Old Lectures 2-3
Lecture 3    Old Lecture 4-6 excluding effects
Lecture 4    Old Lectures 7-8
Lecture 5    Old Lecture 9-10
Lecture 6    Trees (11)
Lecture 8    Local Definitions and Lexical Scope (18)
Lecture 9    Function Abstraction: Functions are Values (16-17)
Lecture 10   Lambda abstractions (Defining functions on the fly) (24)
Lecture 11   Accumulators and tail recursion (26-27)
Lecture 12   Generative recursion
...
```

Access Control: (Please don't edit)

- Set ALLOWTOPICCHANGE = Main.CSFacultyGroup