I. Course Introduction
   A. Compilers are omnipresent
      1. You don’t use a language without some sort of compiler
   B. Compilers have and continue to become more important
      1. As architectures get more complex
         a. Optimization for them becomes more difficult
         b. Often compilers can do better than humans
      2. Compilers can do things like automatic parallelization
         a. Take advantage of multiple cores
      3. Most of the changes to Java are in the compiler, not run-time
   C. Compilers are one type of tool that analyzes programs
      1. Working from source or other representation
      2. Get an understanding of what the program does
         a. Doing some sort of analysis
      3. Use this understanding for particular tasks
         a. Code generation
         b. Optimization
         c. Error checking
         d. Type analysis
   D. Program analysis tools in general
      1. Closely related to compilers
      2. Are also of increasing importance
         a. Security checking (MS)
         b. Type safety
         c. Safety property checking
      3. Are key components of other tools
         a. Software Visualization
            i. Provide program-specific visualizations of thread/tasks/transactions
               ➢ Need to understand what are tasks/transactions
Need to understand how to instrument the code
   ii Efficient instrumentation
b. Code search (S6)
   i Understand code fragments semantically
   ii Manipulate the code fragments (transforms)
   iii Resolve the code fragments
c. Programming environments
   i Eclipse quick fix
   ii Eclipse refactorings
   iii SEEDE continuous execution
d. Program verification
   i Abstracting code into a finite state model
E. Lots of different program analyses
   1. Use-def linkages, Reaching definitions
   2. Escape analysis
   3. Data flow analysis
   4. Type analysis
   5. Loop optimizations
   6. Parallelization
   7. Shape analysis
   8. Alias analysis
   9. Information flow (security)
F. Compilers and program analysis are closely related
   1. Compilers depend on program analysis
   2. Program analysis depends on compilers
   3. They use the same representations and techniques

II. Course Mechanics
A. I want to cover both compilers and program analysis
   1. With an emphasis based on class interests
   2. Therefore I’m going to need feedback now and throughout the course
   3. We have an on-line questionnaire that should be filled out by Monday
B. First issue is when to meet :: decide this next Tuesday when we have a better sense of enrollments
   1. This time works for me.
   2. I can also meet Monday 3-5 if that is more convenient for everyone
   3. We can make this effectively a group-independent study if it remains small and interests are shared
4. Other alternatives are also possible

C. Second issue is grading
   1. Much of the course will be project based (75%)
   2. Assume there will be a take-home final (10%)

D. Text book
   1. *Engineering a Compiler* (Second Edition) by Cooper and Torczon
      a. An on-line version (pdf) is available
   2. We will pretty much go through the text in order
      a. Supplementing with other material, especially for program analysis
      b. How much time we spend will depend on interests
   3. Will assign some problems from the text to help with final
      a. These will be collected and part of the grade

E. Assignment for Tuesday: Read chapter 1

III. Projects

A. Projects can be individual or 1-2 person teams

B. The project should be something of interest to you
   1. It can be writing a compiler for a simple language.
      a. The Appel books include a simple functional-style language
      b. I have a simple object-oriented language called Decaf that illustrates
         many of the principles of object-oriented compilation.
      c. Can also look at simple parallel languages for graphics processors
   2. It can be writing a detailed program analysis for a real language
      a. For example doing type inference in Python
      b. For example doing escape analysis in Java
      c. For example, doing a security analysis of a web back end
   3. It can be writing an application that makes use of one or more program
      analysis
      a. Software visualization of use-def chains
      b. Automatic discovery and display of clones

C. Assignment for Tuesday 9/12
   1. Determine preferences for meeting modes and times
   2. Choose a perspective project or project area
      a. Understand what the source language is going to be
      b. Know what language you want to implement your project in
      c. Be prepared to describe your perspective project to the class
   3. Note this is not a firm commitment, just an expression of interest
      a. Will help me better organize the course
b. Will let me provide feedback as to feasibility and approaches

c. Will let you recruit partners or decide to work with someone else