When students have completed CS61A, *Structure and Interpretation of Computer Programs*, we expect them to be able to:

1) Write programs in functional style, including recursive functions and higher order functions.

2) Write programs in object oriented style, including class and instance variables, and inheritance.

3) Write /basic/ programs in declarative style, including recursive relations, although we don't expect real proficiency in this style.

4) Compare and contrast functional, object oriented, and declarative programming, and select the best style for a given project.

5) Use data abstraction in programming projects.

6) Write programs using hierarchical (recursive) data structures such as trees.

7) Use lazy/delayed/normal-order evaluation to write programs using lazy lists, including infinite lists.

8) Write an interpreter for a syntactically simple "little language" such as might be used as the extension language for an application.

9) Define and use the concepts of order of growth, recursive vs. iterative process, normal vs. applicative order evaluation, client-server programming, dynamic vs. lexical scope, binding, environment, mutation, continuation passing, concurrency and serialization, and linked list vs. vector.