Study Guide

COS 441
Princeton University
Fall 2004

The Final

30% of your final grade
Available 1/3/05
Due 1/11/05 5pm (Dean’s Date)
A about as long as the midterm. Same
ground rules (i.e. open books, notes…)
(Should not take you seven days!)
I have not written the test yet, but will use
what I present here as a guide.

Material Covered

• Assumes you understand all the material
covered on the midterm!
• Covers topics from Week 5 – Week 12 on
the schedule (not including Week 6)

Advanced Control Features

• Understand op-semantics for
continuations and exceptions
• Be able to correctly reduce an expression
that uses call-cc or contains exceptions
• Understand the difference between
implementations techniques for
continuations and exceptions

Denotational Semantics

• Understand how a DS is different from op-
sem
• Prove equivalence of DS equations
• Derive a non-standard DS for a language
(section 4.3.4 in Mitchell)
• Understand DS for while-loop like
language in Mitchell 4.3.3

Types and Modularity

• Understand typing rules and op-sem for all
the related constructs (pairs, sum, rec, …)
• Be able to translate and ML datatype into
primitive pairs, sums, and rec operators
• Concepts of modularity introduced by
Mitchell component, interface,
specification, implementation, constructor,
destructor, operator…
<table>
<thead>
<tr>
<th>Polymorphism and Subtyping</th>
<th>Object-Oriented Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Understand op-sem and typing rules</td>
<td>• Key concepts introduced by Mitchell</td>
</tr>
<tr>
<td>• Be able to complete type-preservation and progress proofs for relevant constructs</td>
<td>– dynamic lookup, encapsulation, inheritance, and subtyping</td>
</tr>
<tr>
<td>• Difference between coercive, inclusive, implicit, explicit</td>
<td>• Relationship between objects and closures</td>
</tr>
<tr>
<td>• Understand subtyping rules similar to Problem 1 on last homework</td>
<td>• Java syntax, and how Java incorporates dynamic lookup, encapsulation, inheritance, and subtyping</td>
</tr>
<tr>
<td>• Relationship between data abstraction and polymorphism</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Java 5</th>
<th>Advanced Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Syntax and Semantics of Featherweight Java</td>
<td>• Nothing specific, but may take examples for other problems from material presented in the advanced topic section</td>
</tr>
<tr>
<td>• Dimension of extensionality</td>
<td>• Familiarity of with material covered here not critical but useful</td>
</tr>
<tr>
<td>• Translating SML like programs into Java using either methods or the Visitor pattern</td>
<td>– e.g. Given and example translation complete it...</td>
</tr>
</tbody>
</table>