

COMP1120-2003-Lab-12

Notes for Laboratory Session 12

1 Topic and Motivation

The aim of this tutorial is to help you prepare for the unit's exam. COMP1120 is a new course, but the final exam will be similar to those for COMP1110. This tutorial will give you the chance to see the sort of questions that might be in the exam; for the format of the paper, see the previous COMP1110 exams.

In order to get the most out of this revision tutorial, you should come prepared. As the exam covers all the material covered in the unit, you really need to revise everything we've done.

2 Preparation

Before attending this tutorial you should (ideally) revise the following from the DA and SE modules:

- Lecture notes from all lectures.
- Laboratory exercises from all labs.
- Tutorial exercises from all tuts.
- Chapters of the textbook referenced in the unit.
- Articles from the Information Pack.
- Assignment work.

3 Tutorial Exercises

1. Write a recursive function to multiply two non-negative integers together using only addition, subtraction and recursive function calls. (Hint: use the formula $m \times n = (m - 1) \times n + n$.)
2. (a) What are the properties of a loop variant?
(b) Give a loop variant for the following code:

```
from  $i := a.lower$  until  $i = a.upper+1$  loop  
     $a.put(2*a@i, i)$   
     $i := i+1$   
end
```

- (c) Give a loop invariant for the code.

3. Design a regular expression to accept a string consisting of an odd number of '1's followed by an even number of '0's.
4. The town of Woolloomooloo has decided to build a new library, with a computer based system for keeping track of the books it holds. As the head software engineer at Woolloomooloo Data Systems, you have been given the job of designing the computer system for the library.

The library will hold two kinds of books: reference books and lending books. Both kinds of books will have an author, title, and year of publication. In addition to these properties, lending books may be borrowed by clients of the library.

- (a) Write a complete class *BOOK* that describes those features common to both reference and lending books. Comments are not required.
- (b) Write an heir of *BOOK* called *LENDING_BOOK*. In addition to the properties inherited from *BOOK*, lending books should support the following features:

```
lend ( client : CLIENT)
    -- Lend to 'client'.
```

```
on_loan : BOOLEAN
    -- Currently out on loan?
```

```
borrower : CLIENT
    -- Who has it?
```

```
return
    -- Return loan.
```

You may assume that the class *CLIENT* has been implemented elsewhere.

Include appropriate **require** and **ensure** clauses in your implementation of these features. Comments are not required.

- (c) When designing an object-oriented system it is often possible to represent some property using either the class hierarchy (the 'is a' relationship), *or* the parts hierarchy (the 'has a' relationship). For example, in the library system just described, books are categorised as either reference or lending books through the class hierarchy. Give an alternative design for this system by defining a general class of books that distinguishes between reference and lending books using the parts hierarchy. You need only give the code for the creation and *lend* features. Include an appropriate **require** clause in your implementation of *lend*.
- (d) Which of these designs—the one based on the class hierarchy, or the one based on the parts hierarchy—is the more appropriate for this application? Why?

4 Laboratory Exercises

None specifically for this week, but there is time for final touches on Assignment 3.