Introduction to Programming
Week 13 tutor’s kit

1. The situation
This week will focus on evaluation, with marking of group demos, individual work for stage 3 and students doing surveys about the course.

It is very important that we get prac work marked so that students get prompt feedback. You must make it a priority to finalise all marks for your class this week. This will enable section leaders to amalgamate them and post them for checking. More importantly, it will mean that your feedback will be given to students right now when they will be most responsive to it.

Note that you will be ensuring that students complete the survey in you lab. You should ask students to do it BEFORE they give the group demo and once they say that they have completed it, they can have a copy of the actual Semester 2 1998 1001 exam. This is the best guide to the style and difficulty of the actual exam.

Note: People will probably ask about week 14 arrangements. As announced in seminars this week, there are no set classes in week 14. The Help Desk will operate. And I will be asking you for tutor feedback and reflection on the semester. I will be trying for a meeting of tutors on Monday at 10am in week 14 ie 7th June.

2. Tutorial script

<table>
<thead>
<tr>
<th>Activity 0</th>
<th>time for it minutes</th>
<th>total time minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview:</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Activities for today are</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Collect class requests for revision</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Review some aspects we know need consolidation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. pause gymnastics (whole class + tutor)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. More revision</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Activity 1</th>
<th>time for it minutes</th>
<th>total time minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect class requests for revision</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ask students to volunteer things to review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Suggest they look at the sample exam in 1001/1901 resource Book</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ask class to help prioritise them</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Activity 2</th>
<th>time for it minutes</th>
<th>total time minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review our problem areas</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select tasks from those listed at the end</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Activity 3</th>
<th>time for it</th>
<th>total time</th>
</tr>
</thead>
</table>
Pause gymnastics
Keep it up - or start now - last chance.

Activity 4

<table>
<thead>
<tr>
<th>Time for it</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>minutes</td>
<td>minutes</td>
</tr>
<tr>
<td>5</td>
<td>65</td>
</tr>
</tbody>
</table>

More revision
You will need to decide between the following:
• more questions from those listed below;
• class questions they want from the sample exam;
• their requests

3. Laboratory script

Overview:
Activities in this week’s lab are:
• Each student does a quiz
• Section leader marks groups demos
• Probably section leader takes posters away to mark
  (depends on whether there is time in the labs)
• You complete seeing individual demos so you can finalise stage 3 mark
  and in the meantime, students might use the self-assessment site
  Encourage them to provide login name and to work seriously
  These will be excellent exam preparation
  Several questions are straight from exams
  Several of the model answers illustrate common student mistakes

Remember some pause gymnastics
Revision activities for week 13:

The areas that students seem to be generally needing help in can be summarised as abstraction. This seems also to be a good time to review group processes if your tute is after the lab (I would leave them to next week if not).

- Kingston p234 Abstraction Q1.2
- Kingston p234 Design Q1-4

• Suppose you had to read in lots of numbers and check that they were each in a suitable range, using code like
  
  ```
  loop
    print("Please enter your age in years: ");
    age := readInt
    endLine
    exit on age <> nil and age >= 2 and age <= 120
    print("Sorry, that was not between 2 and 120; please try again
    end loop
  
  If there were four such numbers to be read (age, height, weight, TER), what would be the best thing to do?

- A student wrote the following code:
  
  ```
  waitKey -> (keypress : String) is
  == wait for user to press a key then return the character
  do
    keypress := terminal.askChar
    keypress := terminal.getChar
  end waitKey
  
  Pause is
  == wait for user to press a key then return character
  var
    keypress : String
  do
    print("Press a key...")
    keypress := waitKey
  end Pause
  ```

The student used these separate routines so that some places the "Press a key" was needed and in others it was not. How would you modify waitKey to do both these tasks? Is this is better or not? Justify your answer.

• How can the following code be improved?
  
  ```
  if age >= 2 then
    if age <= 120 then
      print("Thank you, your age is ", age, ", age, \\
    else
      print("I’m sorry, your age can be at most 120
    end if
  else
    print("I’m sorry, your age must be at least 2
  end if
  ```
• Write a routine that can print an LList of integers separated by commas, as in

25, 18, 9, 6

If the list is empty there are to be no numbers and no commas.

• Write a routine that removes every second element of a given LList

• Kingston p217 Basic constructs (cont) Q5 (some idiomatic code) and 7 (to sell enumeration classes)

• Kingston page 222 Q33 shows set expressions - if they do not know if these are legal here is a chance to sell the syntax diagrams at the back of the book.

• page 226, actually starting on page before Q 12, 13, 14 about carefully reading code and code cliches and writing it

• Self assessment questions on groups: Kingston page 233 lower

• Suppose Salary is an LList of integers, print the smallest integer in the list

• Now redo this task, with Salary an array of integers

• trickier task: using the same Salary : LList, print the second largest value.

Judy Kay
1999 PBL Co-ordinator for Comp1001/1901
June 1999

PS. Notes on sample exam in Resource book and part of last week’s tute work:

Q1.i is pretty nasty as you have to think about the possibility of the parameter being zero or negative. Actual exam should not be this nasty but as a tute task, it invites talking about actual intent and taking acre to think of full range of values of an integer.

Q1.ii is also pretty silly since the inner loop only executes the first time through the outer loop. Does require careful reading past seeming intent to actual code action. Use this to discuss the way to fix it - it is a classic student bug.

Q6. iv refers to "this" class (from part iii) so they should do it for both classes. Similarly part v. And just ignore vi.