Introduction to Programming

Week 5 tutor’s kit

Themes

Prototypes, planning; code cliches (traversal, linear search) simplification in design; brainstorming revisited for starting Problem 2; heuristics for testing; self-assessment tasks for control flow aspects of Blue; designing good tests intuitively for small code segments; quiz.

1. Notes about environment and situational issues for this class

• This is the beginning of Problem 2 and a turning point in the semester. You should use this an as opportunity to get students solidly on course. This is the time to say (again) that Problem 1 was a warm up - and many people learnt a lot but now we get to the real thing.

• One important new element in Problem 2 is that they will be assessed on the basis of group and individual plans and progress data. See the workshop comments for more information on this.

• There will be a quiz every week from here on. Note that it is not actually part of the assessment but you should keep a record of their marks. If anyone asks, (and they are bound to) this is so you know how they are going. Your section leader will ask for a summary of how the class is doing on the quizzes. Even though it is not worth marks, if you take it seriously, they will too. Your attitude to things is very important.

• Students must work 6 hours per week every week from here on to get full benefit from the course. They must expect to book terminals and use our machine for about 2 hours per week (though ideally they will work with others in that time)

• People who cannot get the Blue CD going should NOT hold out for some miracle that comes with it. They must start working now. Also at the current stage, they are well advised to try to do work with other students rather than alone as it is still early days and very easy to get stuck. At the same time, there will be more support after Easter.

• If you have some people who are behind or seem to be doing little outside class, you must make sure you see them this week and discuss the matter. You can decide how to do this - perhaps let the bulk of the class finish 5 minutes early and tell a few nominated people you want them to stay back after the others leave. Or (better I think) you might do it one to one during the tute or lab. You can then tell them firmly but politely, that you ‘are worried about them’ - if you do this one to one, you can ask them to tell you about any problems they have that have prevented them from progressing. Let them see you write down notes on what they say so you can come back to them next week about it. People who are behind now should be able to recover but they have to get going on things.

• These notes are written as though the Tute was before the Workshop. If this is not the case for your class, please do the group formation for Problem 2 in the Workshop and your review of the planning in the Tute.

2. Tutorial script

<table>
<thead>
<tr>
<th>Activity 1</th>
<th>time for it minutes</th>
<th>total time minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview class:</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Main activities are:
• introduce the new - more serious - real COMP1001/1901 and please take care to mention the points on the first page
• quiz on loops and nested loops
• group formation activity
• testing activity

<table>
<thead>
<tr>
<th>Activity 2</th>
<th>time for it minutes</th>
<th>total time minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quiz for this week</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The tasks are at the end of these notes as Attachment 1. You decide which ones you want to set your class. Select at least one from set 1 and one from set 2. If everyone struggled on the first one, you could stop them after it and go through the answer and see if they can then work out another similar level one.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1 Ask the students to work individually, but open book on as many of the problems you offer as they can in 15 minutes. <strong>Remind them this is open book - like our exams</strong></td>
<td>15</td>
<td>25</td>
</tr>
<tr>
<td>3.2 Ask them to swap answers and mark them as you go through answers on the board - or if you have walked around during the class, pick a student who did a good solution to put theirs on the board</td>
<td>20</td>
<td>45</td>
</tr>
</tbody>
</table>

The marking is 0/1/2 with
2 meaning that all the logic is correct, and the style is ok
1 meaning some good structure but a serious flaw
0 meaning it is very seriously flawed

Students will learn from reading each other’s attempts.
Please collect the marks for your students.

Note that if you have the time, get the class to do more questions.

<table>
<thead>
<tr>
<th>Activity 3</th>
<th>time for it minutes</th>
<th>total time minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group formation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Remember that students cannot start Problem 2 until they have earned certification for Problem 1. All can still participate in this exercise on the assumption that they will all achieve this very soon. At the same time, you need to make sure that there are no more than 2 non-certified students notionally attached to a group.

*** Note that this might be deferred to the lab session if it comes after the tute - in that case you can have more time for more quiz questions. |                     |                    |
| 3.1 Explain that they about to form groups. Explain that groups should be **diverse**. | 10                  | 55                 |
| 3.2 Each student write a personal profile
This is about a page and should be used in the next stage. | 15                  | 60                 |
| 3.3 Groups actually form
Ask them to look at each other’s profiles and use this to form diverse groups.
Warn them that this will not be as comfortable initially as homogeneous groups but it will make groups more effective for all members. On the matter of accreditation on Problem 1, it would be ideal if each group formed had two people who have earned accreditation.
Note that the choice of problem for Problem 2 is part of the decision of group formation. | 15                  | 75                 |
You will need to be very firm then as you go around and see that they have played the game. If you get, say, all the programming experts in one groups, you can very nicely tell them that none of the other groups has any (or as many) programming experts and they should consider moving. You might have to be a interventionist here. Groups should ideally have 5 members.

<table>
<thead>
<tr>
<th>Activity 4</th>
<th>time for it minutes</th>
<th>total time minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discussions on testing</td>
<td>35</td>
<td>110</td>
</tr>
</tbody>
</table>

The students should do:
• doing question 1 on page 117.
• Kingston page 234-5 Questions 1 and 9.
You have now had enough experience to decide how to run this, with students working in groups to start with and then bringing the full class together to summarise.

3. 3-hr Workshop

This is the beginning of what will become the routine for the semester in that you will spend 1 hour in the lab reviewing the plans for each person and for each group.

In the group plans, you would expect to see the students list and share out some of the following jobs:

• explore the domain of the problem, learning more about the task at hand
• explore the marking scheme, identifying those aspect that need to be learnt for the first (fast approaching) deadline.
• learn about prototypes (they should see that this is on the schedule for the week 5 seminar) but should also look in Kingston and could do web searches or find books in the library
• learn about testing - also from Kingston and the web and books - the Kingston material refers to a book by McConnell and he has a very fine web site which has check lists for prototyping and for testing and the like.
• have a brainstorming session on how they would like to tackle the problem and its various parts

You need to emphasise the importance of the marks for ‘Reflective diary, plan and progress data’. You will need to clarify what this means:

• each week you will expect each group to provide a plan - this is your chance to help smooth the progress of each group and to head off problems - if the students take it seriously, you can really facilitate their progress. You should mark off each group’s plan on a mark sheet and keep a copy of it so you will be able to refer to it next week. This will be done this week and every week.
• starting next week, you will also expect to see a summary of how each group has progressed on the week’s plan, what was achieved and, for aspects that were not achieved, an explanation and plan for action.
• each individual should provide a plan which, above all, must state the deliverables that they will produce in the coming week. Each individual should have some work that will contribute to the group’s progress. In addition, each individual MUST map out a piece of code they will write in order to build up their skills in Blue. The student should decide what that code should be. If they cannot think of anything, here are some suggestions:
  • some of the tasks at the self-assessment web site
  • code up some examples from Kingston to get more confidence in some aspects of Blue
• code up some small routine which enables them to explore some aspect of Blue that they feel insecure about (perhaps I/O or code that does something they had only barely understood in Problem 1.) You might call for suggestions from the class and put these on the board so that those who need inspiration can get more ideas.

• emphasise that next week, they should bring in some code they will have written this week - An essential deliverable for EVERY week is some code written that week, with comments explaining the motivation for writing that code (eg to get practice in LLists) and the code should be at least a half page of non-tedious code (ie a half page of prints is not acceptable).

Marking of Problem 1:

• You will also need to spend some time this week marking Problem 1 solutions. Use the workshop time to see the 5 minute demos. Try to see all of those that are ready to mark. (I write the time they start on the marksheet and make it clear that I will stop them at around 5 minutes - this is important as the demos can drag on and you won’t see them all - and they need to know how to keep to time limits.)

• Please try to get as much marking done as possible this week. It would be wonderful to finalise the marks before the end of Easter.

• Note that you should be generous in the marking. It is pretty discouraging for a student to work really hard to get the work done and then have someone nitpicking on details and being really tough. The criteria were written to be achievable - they get somewhat tougher for Problem 2. But for Problem 1, a serious attempt by a student who has taken note of the marking criteria should get very good marks. I found myself giving little ticks for criteria that were partly fulfilled and big ones for most.

• Although you should not be mean with the marks, please save the full marks for just those students who really fulfill all the criteria listed - and I do not expect many of these.

• Note: You (and the students) may wish to know that we review each tute group’s marks at the end of the semester to see if they should be scaled. We look at the prac marks for each class and the exam performance for that class. So we can correct for large differences in markers. However, it is far preferable to have a good level of consistency - please feel free to discuss the marking with your section leader - you might take a couple of student’s work and discuss how each of you in the section would mark that level of solution.

Remaining things to remember:

• Remember the pause gymnastics!!!

• And make sure that you encourage the students to take a break during the 3 hours - and to leave the lab for a few minutes.

• AND PLEASE IMPLEMENT the new policy - no food in the labs.

• Finally, remember to mail feedback, comments and requests for seminar content to your section leader.

Judy Kay
1999 PBL Co-ordinator for Comp1001/1901
March 1999
Attachment 1

Problems to use for the Week 4 quiz

Set 1:

Write a routine which does the following task:

i) Given N: Integer as a parameter, the routine prints the numbers N .. 1, one per line. eg with N = 3, it prints

3
2
1

ii) Given N: Integer as a parameter, the routine prints the numbers 1 .. N, with 7 per line (except the last line which may not be full), eg with N = 11, it prints

1 2 3 4 5 6 7
8 9 10 11

iii) Given N: Integer as a parameter, the routine prints the numbers 1 .. N, with their square and then square root, eg given N = 4, it prints

1 1 1
2 4 1.41421
3 9 1.73205
4 16 2

(Note sqrt is not easy to find in the resources as it is not in an index but if they have read Kingston they should have seen it heaps - it is also in the Real class interface p67 of Blue Ref)

iv) Given N: Integer and an LList<Integer>, it returns the number of times N occurred in the the list, and with the list of numbers 1,2,3,1,2,1, and with N=1, it returns 3.

Set 2:

i) Given N: Integer as a parameter, the routine prints one line with N 1’s, one line with N 2’s ... up to a line with N N’s, eg with N = 3, it prints

1 1 1
2 2 2
3 3 3

ii) Given N: Integer as a parameter, the routine prints a similar pattern to above but with columns eg with N = 3, it prints

1 2 3
1 2 3
1 2 3

iii) Given N: Integer as a parameter, the routine prints the lower triangle of the above, eg with N = 3, it prints

1
1 2
1 2 3