

Iteration 2 Topic 7

Topic: Introduction to polymorphism

Attendees: 7

Summary

OB1: The teacher commences the lesson by screen-sharing the polymorphism program through his IDE, running the program, and discussing various aspects of the program (using audio). Following this the four files in the program (interface, main method, and two classes implementing the interface) have been placed on the whiteboard so that students can relate the features of the code that reference between files, reducing the split attention (Ayres & Sweller, 2005) experienced in the previous iteration. This is shown in Figure 94.

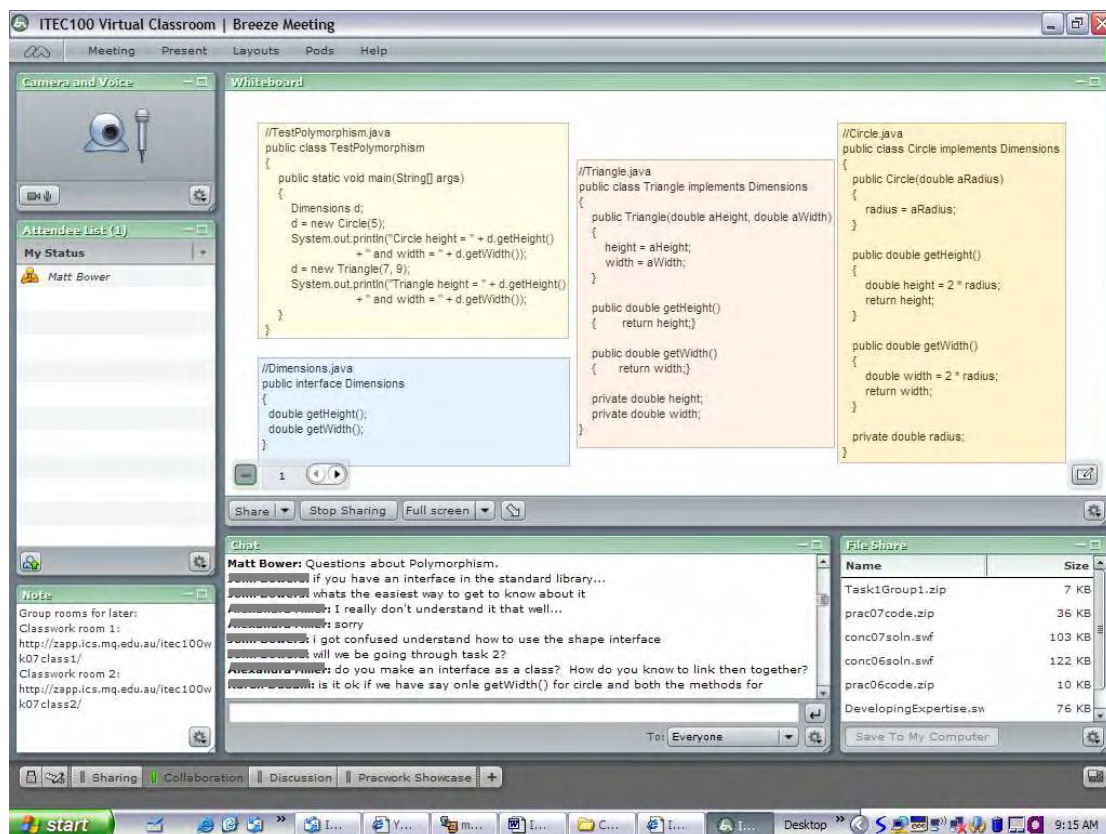


Figure 94 – Iteration 2 Topic 7 Using the whiteboard to present several source code files at once

After having spent several minutes using a purely transmissive approach to describing polymorphism, the teacher asks whether students have any questions. The student text-chat discourse that results includes:

- MB: Questions about Polymorphism.
- JB: if you have an interface in the standard library...
- JB: whats the easiest way to get to know about it
- AM: I really don't understand it that well...
- AM: sorry
- JB: i got confused understand how to use the shape interface

JB: will we be going through task 2?
 AM: do you make an interface as a class? How do you know to link then together?
 KD: is it ok if we have say `getWidth()` for circle and both the methods for triangle?
 JB: you would need to have an empty method in the realisation
 GV: yes i think that would be ok
 JB: you would need to define both methods in any class that realises an interface
 XS: what does that mean? only implement `getWidth()` in circle?
 JB: but one of the methods may not have any implementation
 GV: ok
 XS: could interface extends interface?
 JB: why would you want to do that?
 XS: o, got it

The approach has resulted in student questions that evidence various levels of understanding from the SOLO Taxonomy (Biggs & Collis, 1982), from prestructural (AM) to multistructural (KD) to relational (XS). Opening up discourse by asking for questions provides the opportunity for the teacher to remediate prestructural understandings, the potential for students to help each other develop their mental models to a relational level of understanding (JB to KD) and the chance for the teacher to assist more capable students (XS) to develop an extended-abstract understanding. There were far more questions under the approach used in this iteration than the previous iteration.

OB2: The preliminary tutorial questions were then covered using the same approach as previous weeks, with students providing feedback to solutions in the other groups' rooms. During the activity the teacher explicitly encourages greater student feedback, which resulted in increased student contributions. The teacher then briefly presented and discussed the model solutions, during which time student contribution rates decline.

OB3: After the break three questions relating to casting are attempted in the main classroom, the first of which is shown below:

Question 1:
Suppose C is a class that realizes the interfaces I and J. Which of the assignments (1, 2, or 3) require a cast?
 $C\ c = \dots;$
 $I\ i = \dots;$
 $J\ j = \dots;$
 $c = i; // 1$
 $j = c; // 2$
 $i = j; // 3$

Through their responses the two most capable students demonstrated a multistructural understanding of when casting was required and were willing to contribute ideas through the text-chat when the teacher asked for responses to each part of the question. Other students were unwilling, and were asked if clarification was required (to which most students responded affirmatively).

OB4: Weakly formed student schema had been anticipated based on the review of previous semester's lesson, and as such a whiteboard diagram that had been prepared prior to the class (see Figure 95). This was so as to leverage the multimedia learning principle (Fletcher & Tobias, 2005) as well as facilitate more efficient instruction by obviating the need to draw any diagrams during the lesson.

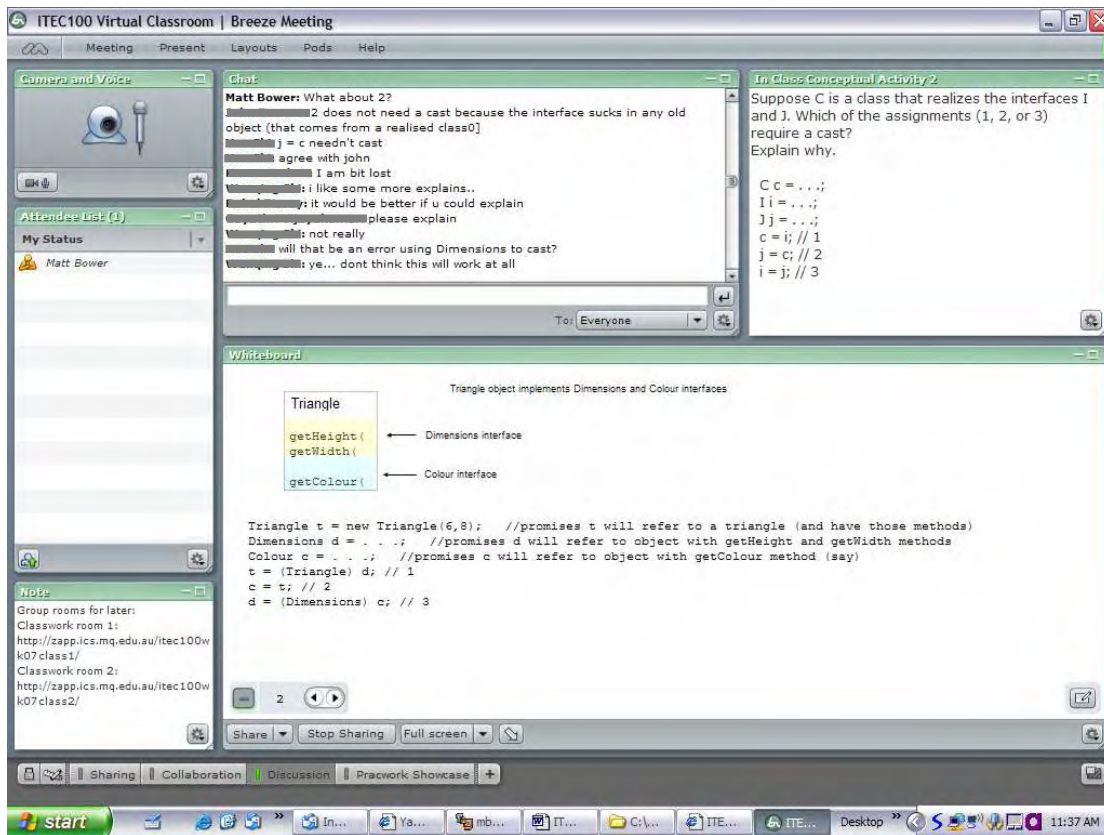


Figure 95 – Iteration 2 Topic 7 Using a pre-prepared whiteboard diagram to support a conceptual explanation

The teacher presented the diagram to students and provided an audio explanation. Students still expressed uncertainty, and to gauge their understanding the teacher presented them with another question to test their knowledge of interfaces:

Question 2:

Suppose C is a class that realizes the interfaces I and J. Which of the following assignments (1, 2 or 3) will throw an exception?

C c = new C();

I i = c; // 1

J j = (J)i; // 2

C d = (C)i; // 3

Responses from the more capable students indicated improved understanding. When students were asked if this had clarified understanding all students responded that it had.

OB5: A final interface question was presented to students:

Question 3

Suppose the class Sandwich implements the Edible Interface. Which of the following assignments are legal?

Sandwich sub = new Sandwich();

```

Edible e = sub; // 1
Rectangle cerealBox = new Rectangle(5, 10, 20, 30);
Edible f = cerealBox; // 2
f = (Edible) cerealBox; // 3
sub = e; // 4
sub = (Sandwich)e; // 5
sub = (Sandwich)cerealBox; // 6

```

This time student responses indicated greater understanding of the concept of interfaces, although some students were still reluctant to contribute their answers. Using real life concepts of “Edible” and the “Sandwich” appeared to facilitate the development of students’ mental models, as did using more than one example.

OB6: Student solutions to the practical exercises (RandomShape and PopulationCounter) were shared by using the note-pod and zip file approach of the previous two weeks. The teacher also broadcast the solutions by screen-sharing his IDE, which allows specific parts of the RandomShape and PopulationCounter programs to be ‘signalled’ (Mayer, 2005b) using the cursor and the programs to be run. A teacher-centred approach to presentation was adopted. The teacher’s explanation resulted in one student expressing several comments of excitement upon finally understanding, however throughout the entire coverage of these exercises students only contributed three text-chat comments discussing conceptual matter.

OB7: The group-work programming activity required students to add a Colour interface to the program used in preliminary practical activity 1 (from the beginning of the lesson). Students were to use a standard screen-sharing layout that had been set up for them in their group-work rooms. This resulted in 137 text-chat contributions from group 1 and 45 text-chat contributions from group 2. While group 1 experienced no difficulty in sharing their screen, group 2 contained no students who had previously shared their screen and struggled to perform this operation. Both groups appeared highly engaged in the task, with group 2 students being reluctant to suspend work when the teacher requested they enter the group 1 room to inspect the solution that their peers had derived. Student feedback at the end of the lesson indicated that students preferred group programming to teacher instruction because it was interesting, fun, and had improved students’ understanding of the material.

Key Incidents:

- KI1: Teacher insistence that people make comments and ask questions during the tutorial question feedback session led to more contributions and greater revelation of students’ mental models. For instance, KD ended up trying to tighten the relationship between the UML classes. RP (and GV) revealed casting object types as a weakness. It should be noted that some students required repeated prompting before they would make any contribution.
- KI2: In the group programming activity Group 1 made rapid progress using the screen-sharing approach. The approach allowed both their mental models and troubleshooting tactics to be revealed. However Group 1 collaborations would have been more effective if the person broadcasting their screen would have also been using audio.
- All three members of Group 2 experienced difficulty sharing their screen which significantly impeded their progress.

- KI3: The pre-prepared diagram did not entirely remedy all students' mental models of interfaces and casting, but all students indicated that it did support their understanding.
- KI4: Four students did not contribute to the before-class preliminary activities this week, which narrowed the opportunity to compare and contrast "group" solutions.

Reflective Notes:

- RN1: Analyzing student solutions to the preliminary tutorial questions creates the opportunity for discussion and clearer formation of mental models. The previous technique of presenting the teacher's solutions first discourages suggestions, as students often take the material as absolute. The process of discursively forming mental model appears more valuable to learning than being presented with the most elaborate and accurate mental model initially. The questions that it raises and the questions that are answered as part of the process appear to promote long lasting and more embodied mental models.
- RN2: That teaching online requires the right pace, right interface design, right balance of instructional and collaborative time, right pitch, right implementation of questioning (providing opportunities, being encouraging). However in terms of forming mental models, it appears different sorts of task structures led to different sorts of collaborations and thus different sorts of thinking occurring. Some of those tasks led to basic thinking reinforcement which can lead to automaticity, while other sorts of tasks can lead to new types thinking skills being developed and clearer mental models being formed, underpinning more advanced, expert thinking. A procedural understanding is based upon knowing "what is". A relational understanding is based upon knowing "why". Both types of tasks are required (to develop both automaticity and more advanced strategic thinking).

Iteration 2 Topic 8

Topic: Introduction to events and control handling

Attendees: 6

Summary

OB1: The lesson commenced with the two students who had audio setup on their machine broadcasting and explaining their two page explanation of Event Handling. Both of these explanations were abstract, not referring to any particular piece of code. Students were asked whether they had any questions after each presentation and they responded in the negative. However when asked if they would also like to see the teacher's explanation they responded that they would. The teacher then used a transmissive approach to present the code for two Applets and ran the programs so that the students could see the output. Subsequent teacher questions resulted in several student contributions, particularly by the more capable students, that served to elaborate the collective understanding of Events.

OB2: The first group-work practical activity was conducted in a purpose built group-work room. The task prescription was provided in a note-pod in the top-left corner of the interface (see Figure 96).

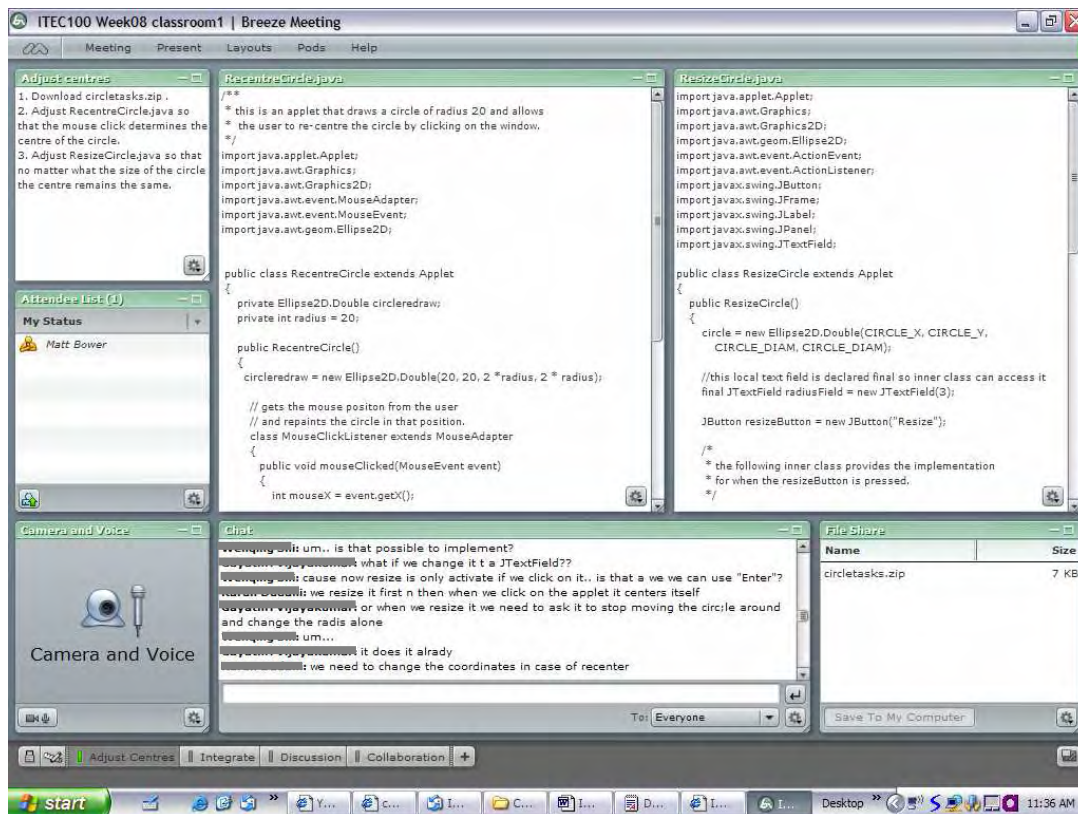


Figure 96 – Iteration 2 Topic 8 Group programming exercise using two note-pods

The task required that students:

1. Download a zip file containing two programs

2. *Adjust the RecentreCircle applet so that the mouse click determined the centre of the circle (and not the top left corner)*
3. *Adjust the ResizeCircle applet so that no matter what the size of the circle the centre remained the same.*

The RecentreCircle and ResizeCircle programs were provided in note-pods and were also downloadable in a zip file. The task prescription was provided in a note-pod in the top-left corner of the interface (see Figure 96). Students collaborated using text-chat and were able to quickly commence and solve the two problems.

OB3: The next task required students to combine the two programs into one applet so that the circle can be both re-centred and resized. For this interface the two previous programs were displayed in note-pods, along with a third note-pod space for the combined program (refer to Figure 97).

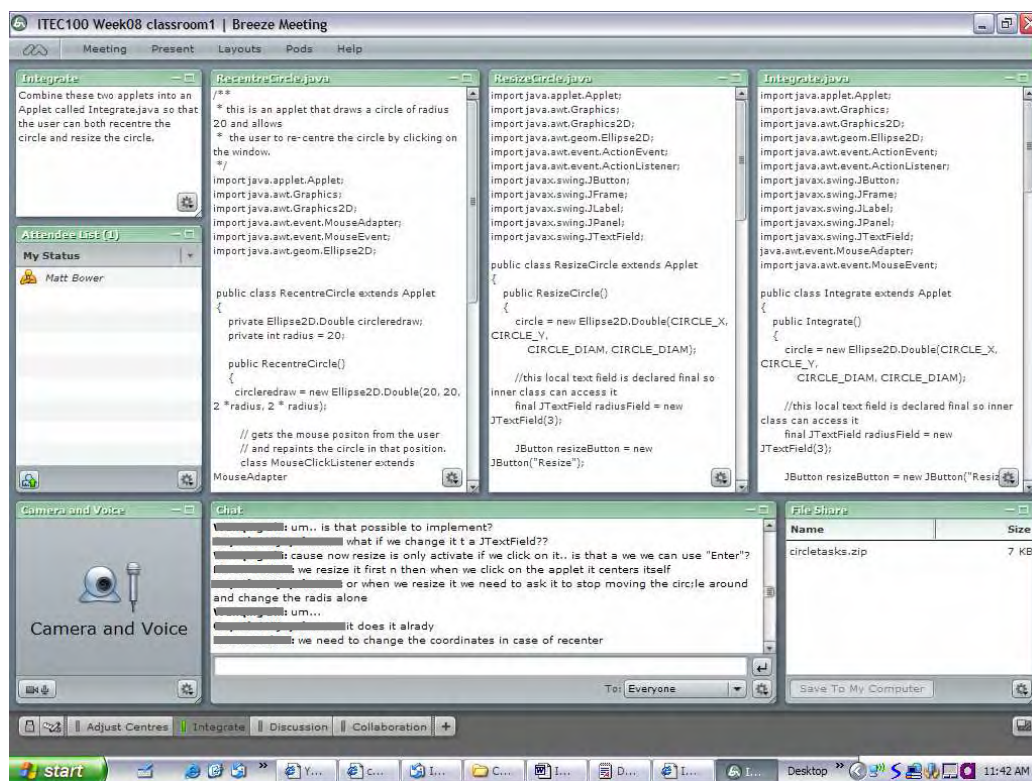


Figure 97 – Iteration 2 Topic 8 Group programming task using a third note-pod

Group 1 was unsure about how to solve this problem. They made some progress, but were uncertain about where the code from RecentreCircle should be inserted into ResizeCircle and did not understand the need to turn the local variables into instance fields. Group 2 were more confident about how to solve the problem and completed the task relatively quickly (compared to group 1 who in the same time did not come close to completing the exercise). In this exercise Group 1 contributed 52 comments as opposed to Group 2 that contributed 78 comments. Note that during this exercise Group 2 also spontaneously decided to maximize the pod containing the integrated program as the other note-pods became obsolete (see Figure 98).

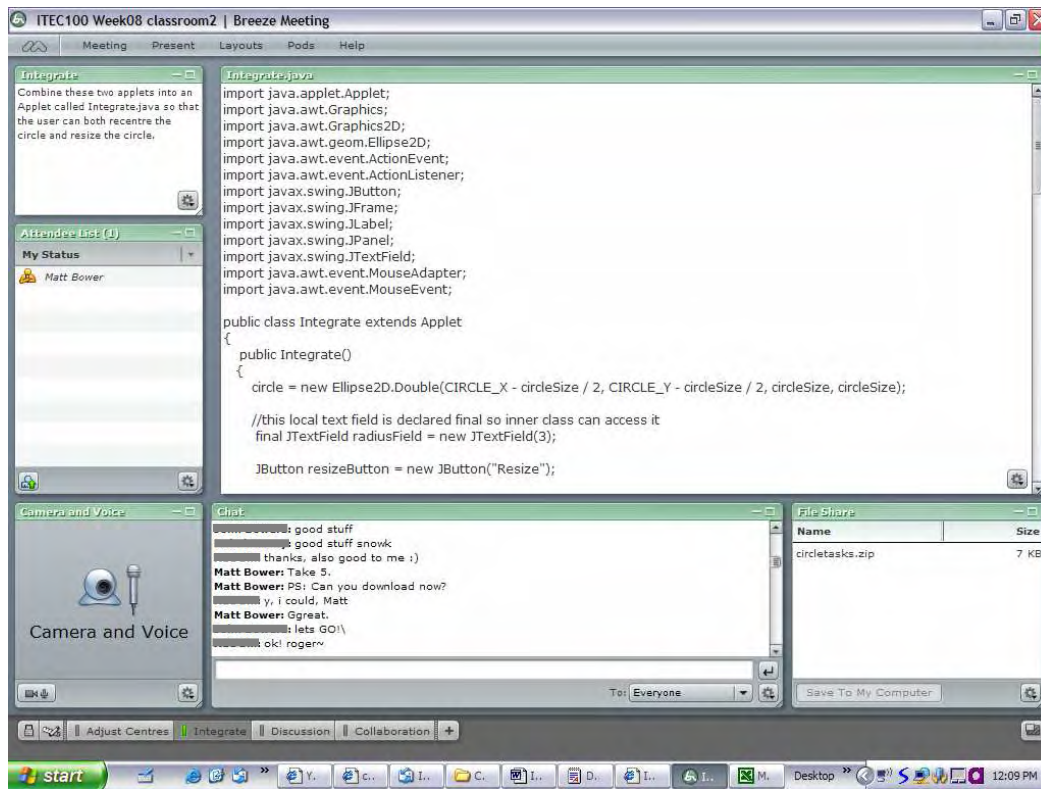


Figure 98 – Iteration 2 Topic 8 Spontaneous student redesign of interface to better meet the changing collaborative needs of episode

This is an example of students spontaneously adjust the interface to suit the changing collaborative and cognitive requirements of the activity (Hollan, Hutchins, & Kirsh, 2000).

OB4: In the final practical activity the teacher used screen-sharing to demonstrate how to adjust a Background Changer applet so that it contains a method that returns a button with an ActionListener attached. The teacher encouraged students to suggest how to solve the problem, and through questioning and prompts tried to achieve a flowing discussion regarding how to construct the method. This resulted in 47 student text-chat contributions during the learning episode, with aspects of Laurillard's (2002) Conversational framework manifesting through the less teacher-centred and more discursive approach. Students commented that completing the activity had supported their understanding of how to make code more efficient by abstracting repeated instructions into methods.

OB5: The final part of the lesson involved the teacher demonstrating to students how to use the wiki for their preliminary conceptual activities in the coming week, including how to add comments and indicate the contributor. The teacher also emphasized the type of contribution that was expected – that even if a question had been answered by other students, individuals could (and were expected to) elaborate.

Key Incidents:

- KI1: It took one of the tutorial question presenters more time than necessary to set-up audio for his presentation because he was unaware that he needed to switch off audio in the main room before he could broadcast in the group-work. As well, the

teacher needed to take microphone off lock in order to avoid an echo sound. However the audio allowed more rapid contribution than possible using text-chat.

- KI2: In the group programming activity the fact that one student was attempting the task on his own machine highlighted version control problems when using the note-pod. Any edits made by other students in the note-pod between the copy and pasting were lost.
- KI3: The fact that the code for the programs in the group programming activity was not able to be contained within the visible portion of the note-pods caused difficulties in following where amendments had been made (i.e., selection difficulties).
- KI4: Valuable discourse resulted from having the team who finished the group programming task offer advice to the other team. This form of review and dialogue was of benefit to the struggling team in terms of the troubleshooting support they received as well as to reviewers by providing them with practise inspecting and debugging code

Reflective Notes:

- RN1: Having students present a training module was a successful means of having them share their mental models. However listening to a broadcast will not result in as tightly developed models as asking students to create a program, because they are not forced to develop a complete mental model when just listening to a description.
- RN2: After XSs Event Handler Training Module presentation no questions resulted, yet teacher enquiry after JB's presentation revealed that some students still did not understand. The more abstract the description and the further removed the description from an actual example, the less opportunity for students to ask specific questions about the concept and create a detailed, complete mental model.
- RN3: Even though only the two most capable students contributing answers to the questions about the code in the teacher's event handling training module, other students are provided with insight into the mental models of these students which provided them with a valuable "vicarious" learning opportunity (Bandura, 1977). It is conjectured that the openness of communication that is adopted for a task (i.e. shared solution space) is as critical as the actual task in developing students' mental models.
- RN4: Transforming the ChangeColourApplet to contain an (efficient) method for creating buttons appeared effective in revealing and developing students' mental models. Students contributed several comments and asked several questions. Programming decision making processes were able to be modelled, offering a form of cognitive apprenticeship (Brown, Collins, & Duguid, 1989). This was manifest through a conversational style between the students and the teacher during the episode. The capacity for specific questions to arise where students might otherwise be stuck if they were working alone, and the capacity to model the abstraction process of constructing a method based upon what was similar and what was different in a sequence of code made this a learning exercise that appeared to be valued by students.
- RN5: It was noted by a student (and the teacher) that the discussion board has been seldom used this semester. The discussion board has proven useful in past semesters for the centralized discursive mechanism it offers, providing an archive that can easily be reviewed. This semester other collaborative means have been used to facilitate student troubleshooting (such as the out-of-class preliminary exercise group-work).

Iteration 2 Topic 9

Topic: Abstract classes

Attendees: 7

Summary

OB1: The lesson begins by reviewing the preliminary tutorial question solutions that students have posted on the wiki. Some students have provided elaborative comments to embellish the contributions of students in their group who contributed earlier. The teacher asks students to load the wiki pages in their browser and provide feedback and comments to the other group's wiki using the text-chat in the main room. Little student feedback is observed. The teacher attempts to provide spontaneous reflections about students' work, however the commentary is noticeably unprepared.

OB2: Students were asked to provide feedback regarding their initial impressions of using the wiki to complete the preliminary conceptual tasks. Students indicated the advantages of the wiki included ability to contribute in their own time, ease of use, capacity to roll-back versions, no overlap of answers (within a group), and ability to see others' answers if experiencing difficulty. Disadvantages included unfamiliarity with how to format, early contributors leaving little that other students can add, and formatting bugs in the tool itself.

OB3: As part of the tutorial questions review and in response to a student question the teacher describes the difference between a shallow copy and a deep copy using the whiteboard to construct a diagram (see Figure 99). This is a time consuming process in part due to the sub-optimal usability of the whiteboard tool. The teacher asks students for their suggestion regarding how a shallow copy is different to a deep copy. However students experience difficulty using the whiteboard so to expedite the process the teacher changes the diagram to dynamically illustrate this difference. The external representation did allow the cognitive load upon students to be reduced as opposed to holding and interrelating all items of information in working memory (van Merriënboer & Ayres, 2005).

OB4: The first practical task required students to download a zip file containing three Square classes that had been designed by class members and then evaluate each (the interface for this episode is shown in Figure 100). Following this they were required to build a Square class in their groups that completely met the specification of preliminary practical task 1 (i.e., by combining the best design features of the three files they had downloaded). It was difficult for students to collaboratively compare and contrast the Square files because the interface provided did not support them in identifying a common point of focus. Students reviewed the code on their IDE without any screen-sharing being used, and because there was no audio it was difficult for them to indicate the lines of code to which they were referring. This was an example of how distributed process loss (Neale, Carroll, & Rosson, 2004) could occur in the web-conferencing environment.

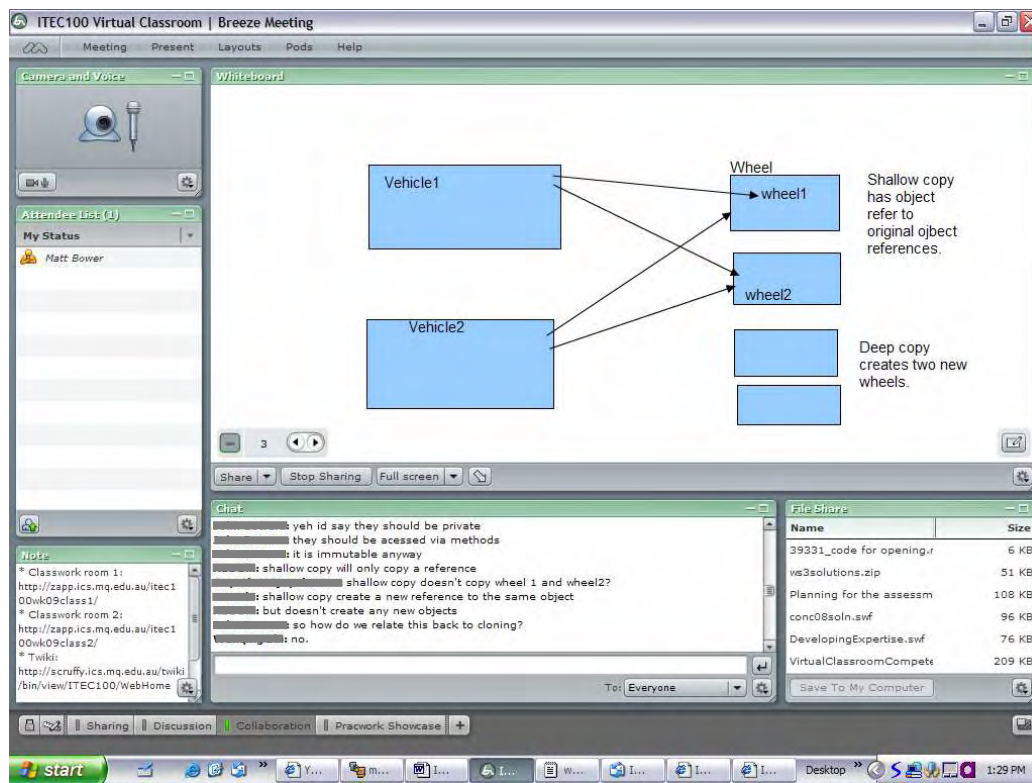


Figure 99 – Iteration 2 Topic 9 Teacher whiteboard diagram to support conceptual explanation

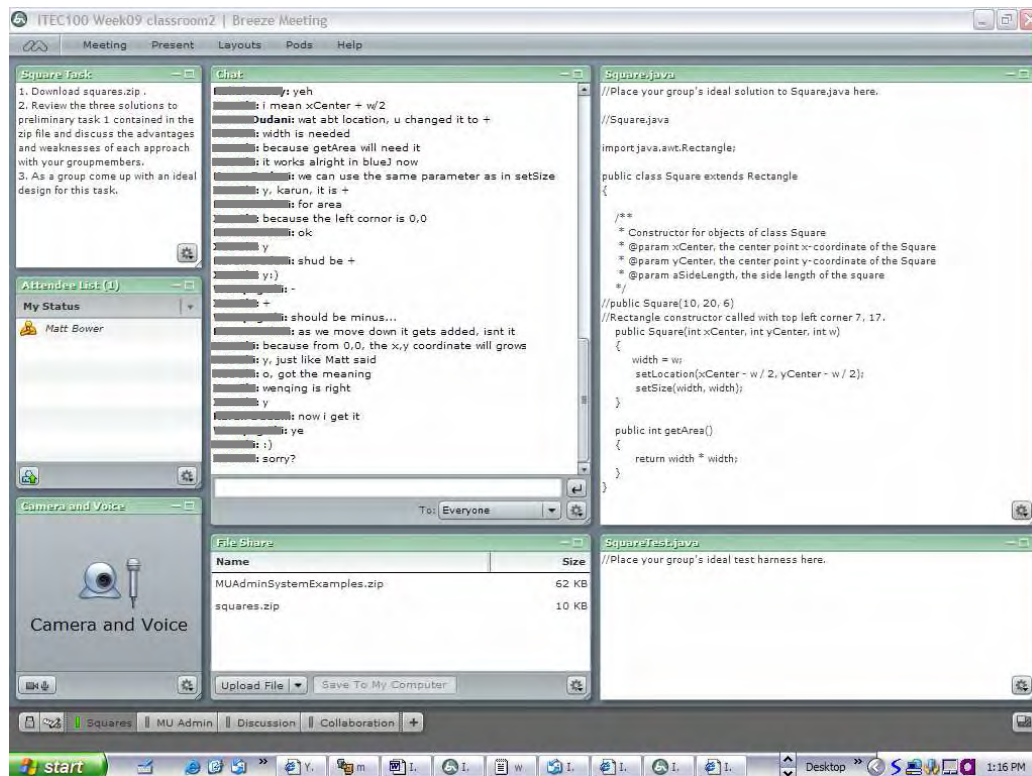


Figure 100 – Iteration 2 Topic 9 Group programming using a note-pod and text-chat

Group 2 made much faster progress because a student copied and pasted the code from one of the supplied programs early, providing a template from which the group could work. Students in both groups experienced difficulty understanding the key issues of:

- not having any instance fields (i.e. using those of the rectangle class via the methods of the rectangle class)
- how to adjust the Square constructor to be the centre of the square (for which the teacher provided an example for Group 2, pointing out that for a Square(10, 20, 6) they would need to construct a Rectangle(7,17,6,6)).

The teacher broadcast scaffolding audio commentary to both groups simultaneously, supporting remediation of these issues. The audio commentary could compliment rather than interfere with student's text-chat and note-pod entries (as according to the modality principle, Low & Sweller, 2005). This task resulted in a great deal of collaboration in both groups, including discussion of semantic/syntactic issues that would most likely not occur in more conceptual tasks.

OB5: Due to time restrictions the teacher demonstrated how to override the toString method for the Square class rather than have the students perform this task collaboratively. This transmissive teacher approach coincided with no student discourse relating to the concepts being addressed.

OB6: Also due to time pressure, the teacher reviewed students' MQ Administration Systems by sharing his screen and providing commentary rather than applying the original plan of having students perform a review and creating an improved system in groups. However because the teacher had not practised this instructional delivery it was unfocused and failed to illuminate key design points that students should heed.

Key Incidents:

- KI1: Discourse during the introductory section requiring students to review the wiki seemed disparate and student comments lacked focus or depth. The teacher's talking seems to dominate collaborations. Whether this is due to or causes the lack of focused comments by students is debatable. The task prescription did not require or support students to focus or sequence their analysis. Students had not yet formed collaborative patterns regarding wiki feedback tasks.
- KI2: Not advising the groups to start with one of the Square classes impeded the progress of one group (an example of how task prescription impacted upon the effectiveness of the learning episode).
- KI3: Once again, using a combination of text-chat and a note-pod for the group programming task resulted in split attention (Ayres & Sweller, 2005) between the two media and failed to capitalize on people's dual processing capabilities (Low & Sweller, 2005). If students had used audio in combination with the text medium utilized in the note-pod they could have collaborated more efficiently.
- KI4: Teacher review and update of the MQ Administration System resulted in no student discussion. Because of the size of the project it was difficult to maintain links between related code in separate files, again resulting in split attention (Ayres & Sweller, 2005). The teacher's transmission based presentation was unprepared and unfocused.

Reflective Notes:

- RN1: Using the wiki as the medium for preliminary tutorial question solutions meant that the teacher did not need to organize group-work rooms with material in them – their work is already and always available online.
- RN2: Attempts to draw diagrams on the whiteboard during class are often frustrating because of the poor interface design of the tool (difficult to select and move objects, no copy-paste, etc).
- RN3: Using a diagram to dynamically illustrate the difference between a shallow copy and a deep copy provides a clearer mental model than a purely audio description (in alignment with the multimedia principle, Fletcher & Tobias, 2005).

Iteration 2 Topic 10

Topic: Graphical User Interfaces

Attendees: 7

Summary

OB1: Because of student difficulties using some features of the wiki (mentioned at the end of the previous lesson) the teacher spent some time at the beginning of the lesson demonstrating how to use the wiki (copying from Word documents, linking to new pages, and escaping CamelCase). Screen-sharing was used to accomplish this (see Figure 101). The teacher also emphasized the value of integrated solutions as opposed to piecemeal solutions on the wiki.

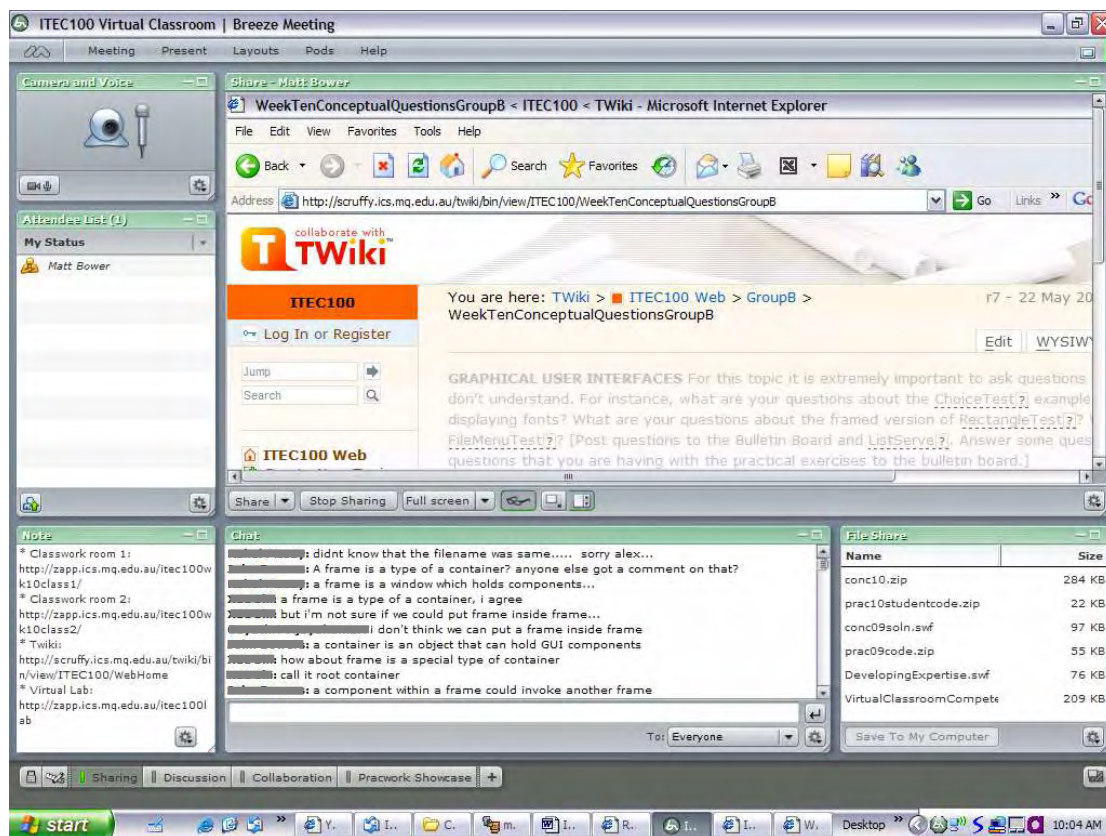


Figure 101 – Iteration 2 Topic 10 Demonstrating wiki use to students using screen-sharing

OB2: The next part of the lesson was dedicated to reviewing students' preliminary conceptual contributions to the wiki. The teacher repeatedly encouraged students to contribute questions and comments however the amount of feedback which students provided was limited. The two most capable students contributed 36 of the 57 comments posted during this section of the lesson, while other students mainly observed their conversation.

OB3: The teacher then reviewed two students' practical task submissions (a RadioButton colour changer and a CheckBox colour changer program) by screen-sharing and running them in the IDE. A question response approach was used to involve students, asking them to explain different pieces of the code as they were highlighted. This comprehension style task was successful in eliciting responsive student participation.

OB4: The collaborative programming task for this week required students to create a program that changes the colour of a panel using a combo-box. The group-work room interfaces had been designed to include the task prescription in a note-pod, an enlarged text-chat pod, and an elongated note-pod to increase the visible area of the program code that could be displayed (see Figure 102).

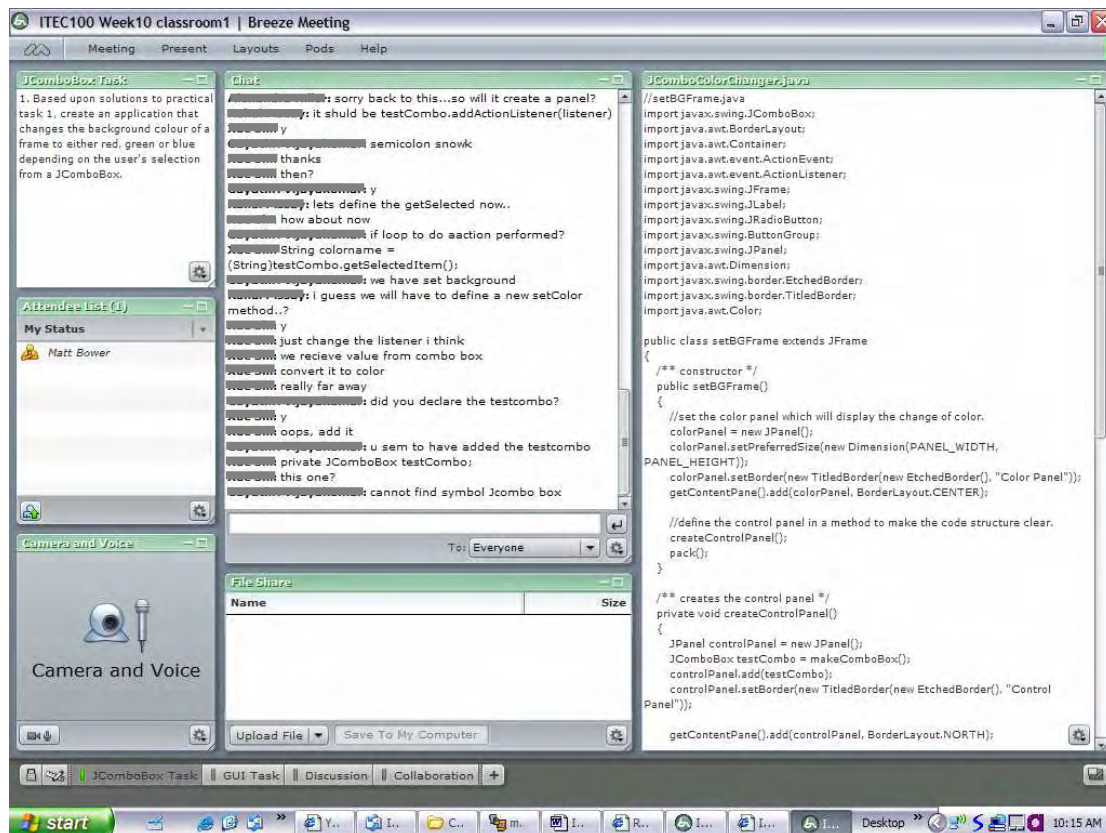


Figure 102 – Iteration 2 Topic 10 Group programming exercise using a note-pod and audio

Initial collaborations in both groups involved a coordinating phase, where students discussed how they would commence the problem solving process (both in terms of coordinating activity between them and how to design their program). Progress was expedited when the teacher advised they use a particular example from the previous practical task review as a starting point for their new program. To complete the task students in Group 1 posted a total of 94 contributions to the text-chat pod, as compared to 34 contributions by Group 2. There were two main reasons for this difference:

- One student in Group 2 assumed a leadership role (asking questions regarding the key programming decisions that needed to be made and performing actions in the solution space) which reduced the need for coordinating discourse.

- The most contributing student (XS) in Group 1 frequently split comments across lines. For instance, in response to a student's question about whether they should define a new setColor method, XS responds:

XS: y
 XS: just change the listener i think
 XS: we recieve value from combo box
 XS: convert it to colour

This “authentic and meaningful” (Herrington, Oliver, & Reeves, 2002) collaborative exercise appeared to result in a great deal of focused and tightly coupled interactions, with many spontaneous questions being posed and many student solutions being offered. Students seemed to be becoming familiar with this approach to group programming.

OB5: Students in both groups were highly active and engaged with the group programming task, and continued working on it even after the teacher had asked that they listen to an explanation regarding the final assignment (which eventuated as the final learning episode for the lesson).

Key Incidents:

- KI1: Notably more synthesis has occurred in the Group 2 wiki-space as opposed to Group 1.
- KI2: A student uploads her conceptual responses to the wiki as an attachment because she was unable to copy and paste her work into the wiki - an example of not having the technological competencies to interact effectively.
- KI3: Changing the ColourChanger program from using radio-buttons to using a combo-box took longer than anticipated, but resulted in substantial amounts of activity and learning oriented discourse. It was difficult to collaborate using the one note-pod because a) only one person can write at a time, b) it is hard to coordinate ideas when communicating through text-chat and the note-pod (split attention effect, Ayres & Sweller, 2005), and c) the scrolling for the note-pod was not synchronized so it was difficult to for students to maintain a shared point of focus.

Reflective Notes:

- RN1: When the teacher requests students contribute questions and comments regarding the wiki task, the unfocused nature of the task prescription appears to have an adverse impact upon collaboration. Students make few contributions and hold little discussion. As well, not having a (visual) point of focus may decrease the quality and quantity of contributions. Note that when the teacher asks students to respond to specific questions during the practical exercises regarding the meaning of code there is much greater student input.

Iteration 2 Topic 11

Topic: Arrays and Streams

Attendees: 6

Summary

OB1: During the review of the preliminary tutorial questions this week the teacher broadcast the wiki using the screen-share facility. Moving around the various student contributions then provided a common focus for students to consider and discuss when reviewing the wiki. As well, the teacher posed specific questions regarding specific aspects of student solutions. In combination, these tactics resulted in greater student involvement and a more focused discussion than occurred in the previous two weeks.

OB2: After completing the tutorial question review, students were asked which approach to completing the preliminary tutorial questions they preferred – submitting individual solutions, using the virtual classroom or using the wiki. A summary of student responses is provided in the following table.

Submission approach	Advantages	Disadvantages
Independent files	Can work in own time Formatting is simple Requires consideration of all questions Quick	Cannot learn from others or progress if stuck Cannot realize own mistakes
Virtual Classroom group-work	Learn from the approach and viewpoint of peers Time restriction encourages faster work Can overcome difficulties and ask one another questions	Requires more time A lot of time may be spent negotiating Requires more technological skill Requires coordinating common time to meet
Wiki contribution	Formatting capabilities useful Can learn from peers Can do in own time Easier to review in class Can use the power of the group to perfect the solution	Formatting capabilities erratic Merging solutions can take time

Table 37 – Iteration 2 student perceptions of contribution approaches

OB3: The next part of the lesson required students to explain how JBs PermutationGenerator program worked. The teacher broadcast the program in the IDE using the screen-sharing facility, and requested that students explain how the code functioned line-by-line. Students were unable to explain convincingly. However the teacher's attempted explanation of how the program operated was confusing because too many pieces of information were being presented in an unorganized manner with no mechanism to support their interrelation.

OB4: The next task required students to describe how a Pascal's triangle program functioned. An interface was provided containing a text-chat pod, the program in an elongated note-pod, the project file for download, and a note-pod containing the following task description:

Download and review (perhaps using the debugger) the prac11studentwork.zip file. Add comments to the constructor for the PascalsTriangle.java file in the task2 program (in the spaces provided) that explain how the program works.

The interface is shown in Figure 103. The previous (teacher-led) practical activity was supposed to model how students should interact on this second (group-work) practical activity. However the teacher had not demonstrated the approach of writing comments to explain how each line operated. This meant that when students began the second practical task requiring them to explain how a student's PascalsTriangle program worked they required repeated prompting to write comments. Students in group 1 struggled to make any progress with this activity, evidently not understanding how the program worked. Students in group 2 only made progress because the program belonged to one of the students in that room (who ended up effectively teaching the other group members how the program worked).

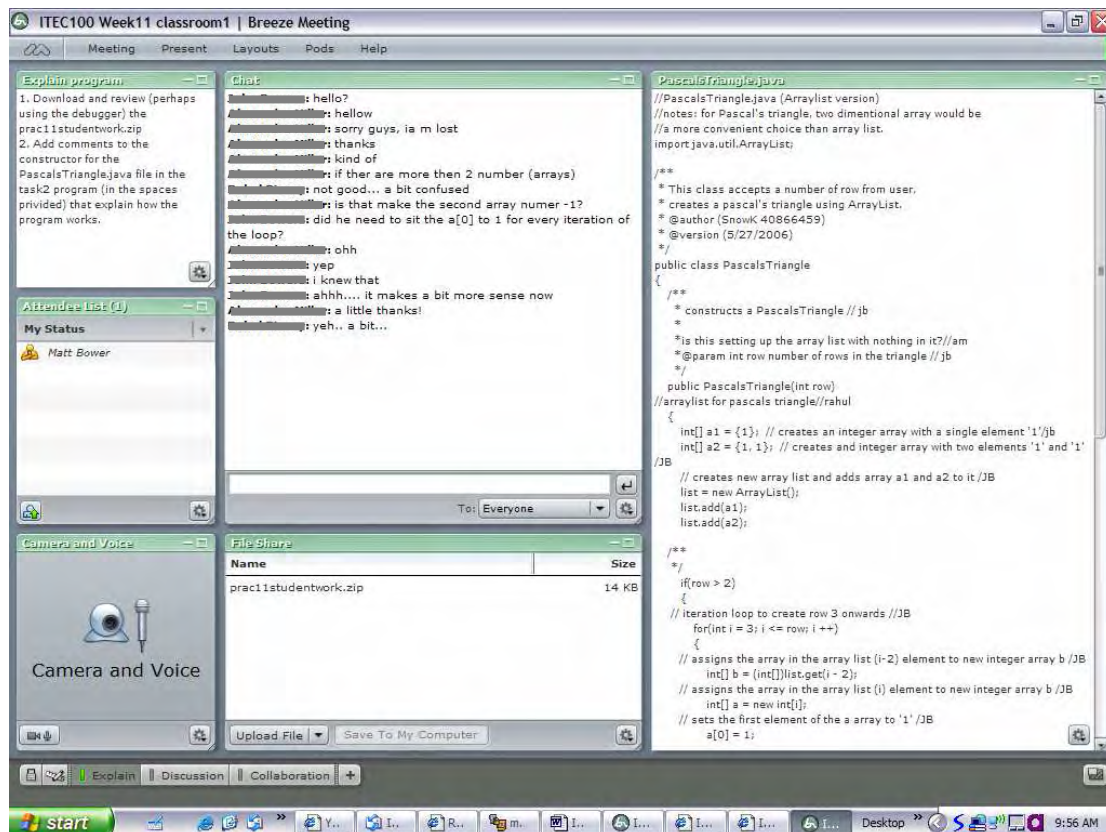


Figure 103 – Iteration 2 Topic 11 Interface for group comment writing task

OB5: Instead of using group-work (as originally planned) for the NumberFormatException debugging exercise, the teacher chose to adopt a teacher-centred approach. The teacher deemed it important for students to have a clear explanation after the confusion they had

experienced in the previous two group programming tasks. Students asked eight questions and the teacher was able to provide clarifying explanations to resolve their misconceptions or points of non-understanding.

OB6: In the final part of the lesson the teacher led a discussion regarding some key concepts for the final assignment, intended to offer students guidance about how the application design could be implemented. Students appeared attentive through this sequence, contributing 29 comments, 20 of which were directly related to responding to or asking content related questions.

Key Incidents:

- KI1: Student feedback regarding which approach to completing the preliminary tutorial questions they preferred was unequivocal. Some people like to just work individually because it makes them work harder and takes less time. Some people like to be able to ask instantaneous questions of their peers. Some people like to learn from others yet complete work in their own time. This points out that there is not one way that will suit all people.
- KI2: The teacher explanation of the student's permutation program was difficult to comprehend because no mechanism was provided to relate the numerous pieces of (audio) information. A notional machine diagram would have dynamically represent how the two arrays were interrelating, providing a concrete example that students could discuss. The teacher's Pascal's Triangle description would have also benefited from diagrammatic support (for instance, via the whiteboard). This would take allow the instructive approach to take advantage of the multimedia principle (Fletcher & Tobias, 2005), and the modality principle (Low & Sweller, 2005).
- KI3: The array practical tasks resulted in low levels of student participation because the difficulty level was beyond most students' ability.
- KI4: The ad-hoc within-task teacher descriptions for the practical activities did not support the development of students' mental models. However the more considered teacher explanation of the NumberFormatException task appeared to address student non-understanding more effectively than the group-work approaches.

Reflective Notes:

- RN1: It is conjectured that the extent of collaboration is an excellent indicator of appropriate pitch. In the first practical group programming activity students in group 2 were able to collaborate because it was at an appropriate pitch for them, whereas students in group room 1 hardly collaborated at all because they struggled with the content matter. Group 2 successfully completed the task with a little time to spare. A teacher can use the extent to which they can engage students in a dialogue (or a dialogue results) as an appropriate measure of pitch.
- RN2: At times, whether an instructive or more student-centred approach is adopted depends on the teacher's perception of student's understanding. For the NumberFormatException task the teacher chose to adopt an instructive approach rather than the previously planned group-work approach because student responses to the previous tasks indicated they were confused. The more directive approach appeared to consolidate student understanding and improve student satisfaction.
- RN3: A student once again points out that the discussion board has not been utilized for quite some time. The fact that students have to proactively access it to check for new postings reduces its effectiveness as a small class collaborative tool because students cannot be guaranteed of activity.

Iteration 2 Topic 12

Topic: Files and System Design

Attendees: 6

Summary

OB1: This week the teacher adopted a more instructive approach to reviewing the two groups' preliminary conceptual wiki work than in the previous week. However students were still involved, posting 78 comments throughout the sequence. The approach of broadcasting the wiki content being discussed was again adopted by the teacher, which provided a common focus for the discourse.

OB2: After reviewing the tutorial questions using a standard question-response approach, the teacher leads an in-class activity requiring students to identify the association, inheritance and dependency relationships in a program. The task is presented in a purpose built interface, with the task description and learning artefact (computer program) provided in note-pods. The text-chat area is enlarged to accommodate high frequency contribution. See Figure 104.

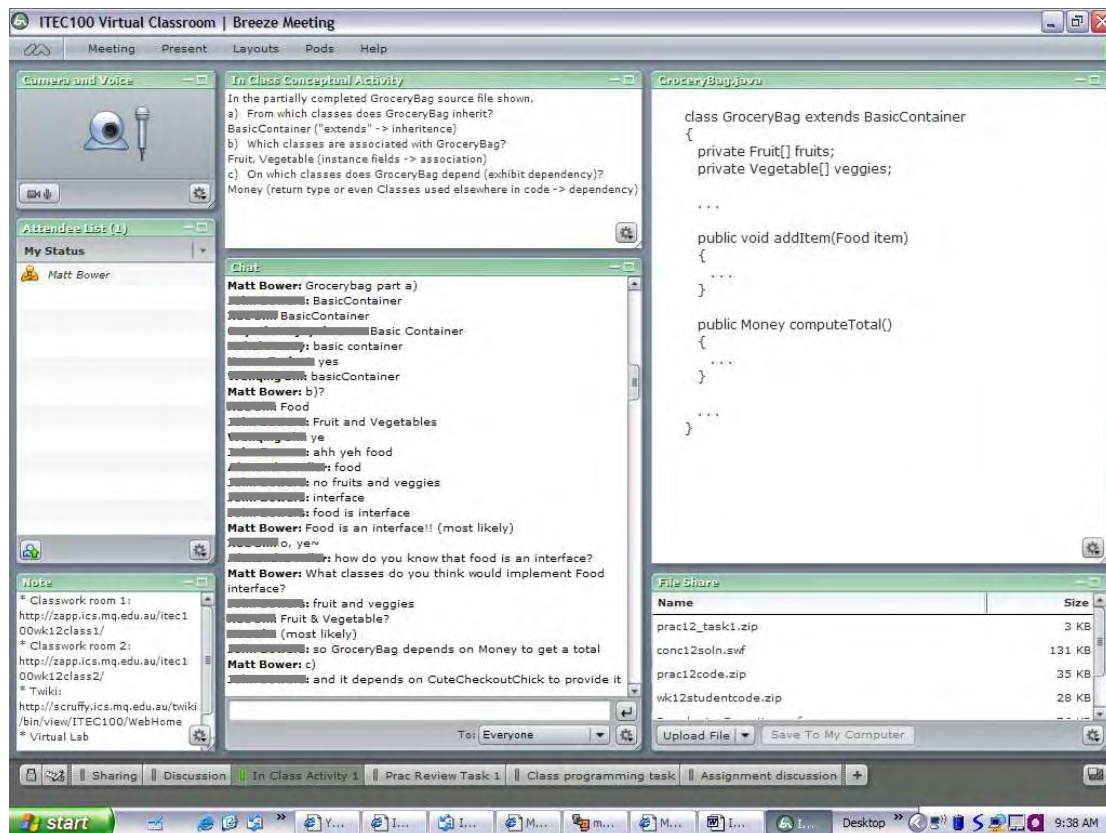


Figure 104 – Iteration 2 Topic 12 Interface for task requiring abstraction

The teacher coordinates activity by prompting students for answers and students contribute 30 text-chat comments. The task challenges students' ability to relate theory to code. For

instance, in response to the teacher question “what’s the difference between dependency and association from the code”, students respond:

JB: if the classes state hold a class object it is association
 AM: location in the programming?
 JB: if it just uses it but it is not an attribute of the class then it is dependency
 XS: association most likely to be instance field?
 JB: which is an IsA relationship
 JB: Instance Variables, HasA
 XS: a return type "object" is a dependency relationship
 JB: a local variable denotes dependence

Student discourse reveals the formedness and accuracy of their mental models as according to the SOLO Taxonomy (Biggs & Collis, 1982). AM conjectures that it could be the position of the variable in the class that determines its relationship, which shows a prestructural understanding of how the principles of association and dependency relate to code. XS confirms that object return types indicate dependency relationships, but is not certain about association relationships, indicating a unistructural understanding. JB describes how to identify the relationship based on the function of the variable in the program and is able to provide elaboration regarding other means of identifying the relationship, demonstrating a multistructural understanding (at least). The task requiring students to identify categories of relationships from the concrete representations provided facilitated performance of the action-process-object abstraction cycle (Aharoni, 2000). The learning design (requiring students to form and articulate their abstractions in a shared space) was critical to revealing students’ mental models.

OB3: For the next activity the teacher asks students to copy their solutions to the first preliminary practical question onto the whiteboard. This facilitates freeform comparison and contrast of student work without causing split attention (Ayres & Sweller, 2005) between program files (see Figure 105). The teacher uses the whiteboard contributions to engage students in a conversation. The teacher and students are able to discuss and evaluate the different approaches all within the one interface. Students posted 33 comments, of which 28 were directly related to discussing the concepts at hand. Examples include:

GV: i converted the char to string
 XS: i compute it by character, while GV did it by string
 XS: think replace mine clause with JB's number formula, it will be a good one
 JB: we should create two constants [like] `private static final ASCII_FULLSTOP = 46;`

OB4: The first group programming task required students to change the basic file-reader program so that it printed out each ASCII character in a file followed by a space and then its ASCII number, one on each line. The group-work room interfaces was purpose built to facilitate this task, with the task provided in a note-pod, an enlarged text-chat area in the centre of the window to facilitate discussion, and an elongated note-pod containing a skeleton program to provide the communal solution space. Note that lengthening the solution space allowed the entire program to be displayed in the visible portion of the note-pod so that edits could be easily detected (see Figure 106).

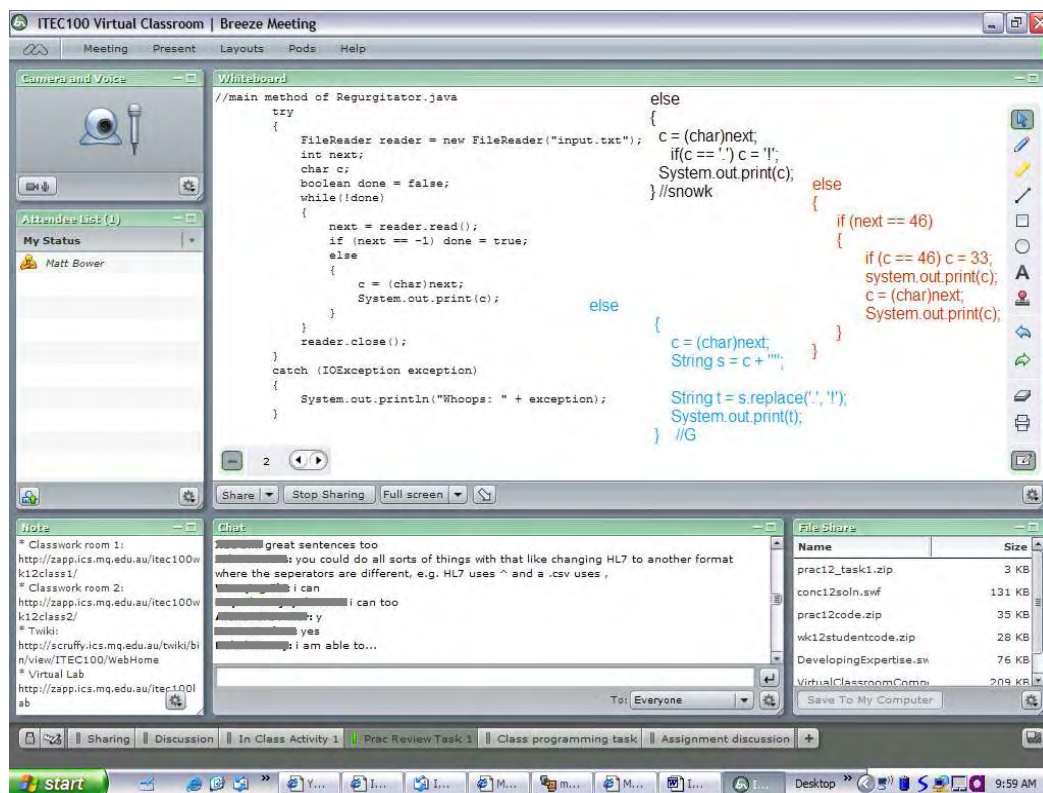


Figure 105 – Iteration 2 Topic 12 Using a whiteboard for students to share pieces of programming code

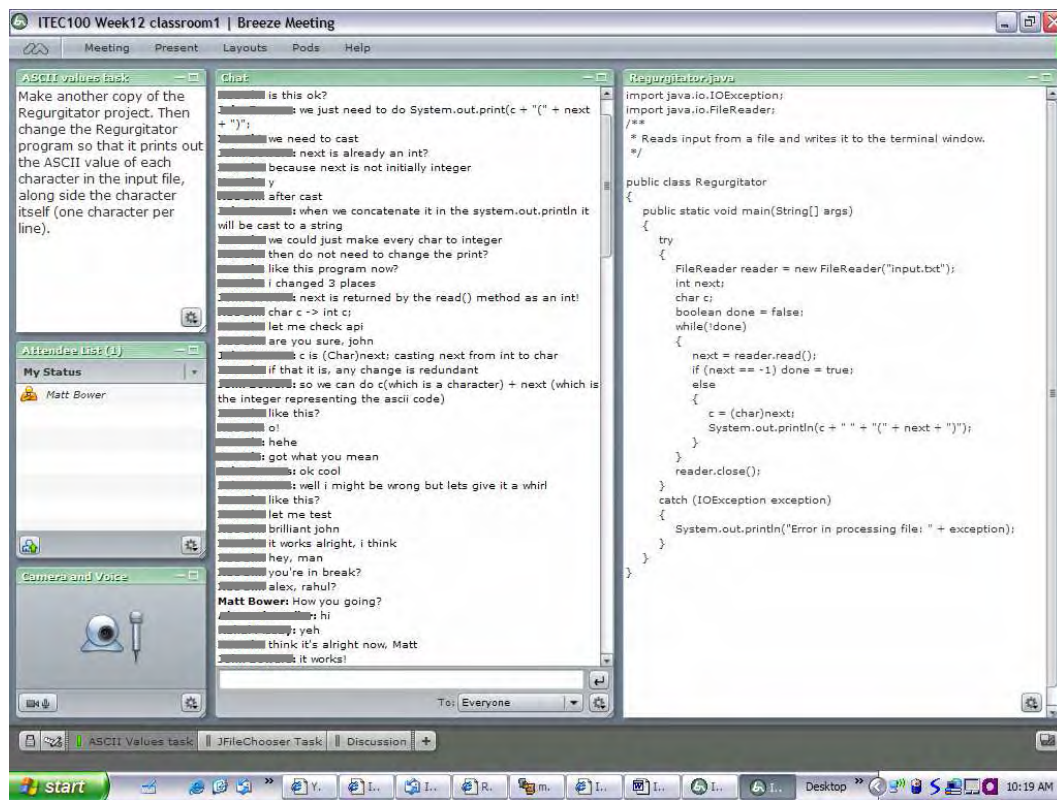


Figure 106 – Iteration 2 Topic 12 A group programming interface using a note-pod

Group 1 completed this in less than two minutes. The two most capable students in the class were both in group 1 and contributed all of the 33 text-chat contributions. The 27 text-chat contributions in group 2 were more evenly distributed between group members. The teacher scaffolded Group 2 efforts with comments and suggestions in order to expedite their attempt.

OB5: After the mid-lesson break students were presented with the group-work task to change the ASCII program from group activity 1 so that it allowed the user to select the input file at runtime using a JFileChooser. Group 1 completed this quickly by having one student share their screen and take (a few) suggestions from other students. Group 2 decided they that they wanted to use a note-pod for to perform the JFileChooser group programming task (Figure 107). When the student who made this interface design decision was asked why, she responded that when she shares her BlueJ she cannot watch others make a comment. She was aware that if she minimizes the browser window they then she receives messages in popup boxes. However she pointed out that this is not the same as being able to reflect upon the chat in her own time. As well, with screen-sharing she cannot contribute back to her peers without audio. This was an example of a student making a design decision to support distributed cognition (Hollan, Hutchins, & Kirsh, 2000) – true use of collaborative spaces to support shared thinking.

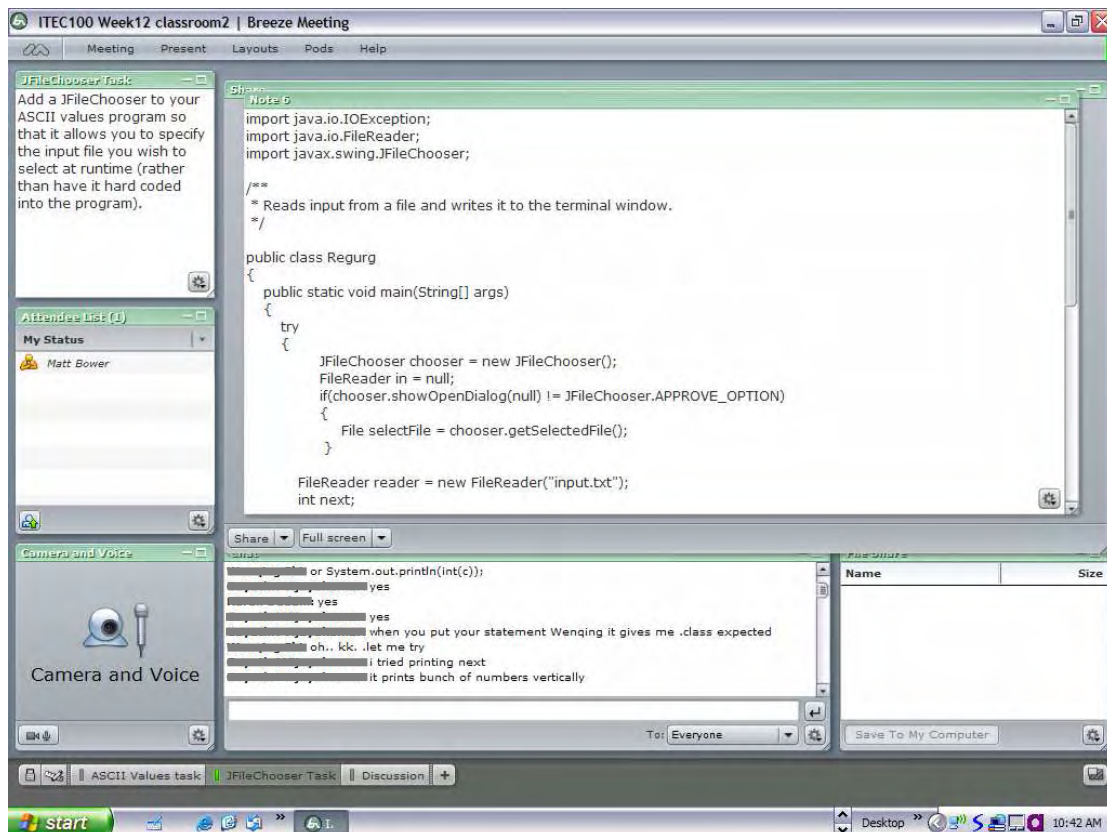


Figure 107 – Iteration 2 Topic 12 Student designed interface for group programming activity

Group 2 were taking longer to complete this task even with teacher support. They were referred to the group 1 solution before they had time to finish.

Key Incidents:

- KI1: On several occasions (such as when covering the wiki work and summarizing the association/inheritance/dependency relationships for the in-class activity) the teacher took a stronger leadership role in the teacher-led activity in order to accelerate the pace of the learning sequence.
- KI2: Amount and specificity of student contributions remained high during the review of the wiki tutorial question work even though the amount of specific questioning by the teacher was reduced. Providing a common focus by screen-sharing the contents of the wiki pages being discussed appeared to be a critical factor in improving collaboration.
- KI3: The GroceryBag in class activity requiring students to link abstractions to concrete examples resulted in discourse that revealed their mental models.
- KI4: The whiteboard afforded the capacity for students to easily contribute their solutions in a manner that preserved formatting. Highlighting (Mayer, 2005b) could be applied using colour to allow solutions to be more easily distinguished. This approach also averted split attention (Ayres & Sweller, 2005) caused when inspecting separate files.
- KI5: By the final week students are making spontaneous and considered virtual classroom collaborative design decisions (for example, deciding to use a note-pod for the group programming task rather than screen-sharing).

Reflective Notes:

- RN1: The difficulties in setting-up student audio resulted in this seldom being used throughout the semester, however this was identified as an impediment to collaboration, especially in circumstances where students wished to perform collaborative programming.
- RN2: The dominant role of the two most capable students (XS and JB) has been a pervasive theme throughout the semester. On most occasions these two students were split between any groups that were formed to facilitate more successful progress within each group.

Summaries of Iteration 3 Lessons – Semester 2 of 2006

Iteration 3 Topic 1

Topic: Introduction to writing, compiling and running java programs, as well as providing a general introduction to the course.

Attendees: 3

Summary of Lesson

OB1: The teacher implemented a similar lesson plan to the first two semesters, to offer a means of calibrating the iterations. First students were provided with a guided tour of the web-conferencing tools, in a similar fashion to previous semesters. In addition, some of the collaborative conventions (such as placing their initials after comments entered in the note-pod) were discussed. From an Activity Theory (Engeström, 1987) perspective, this was establishing rules for the community from the outset. Student audio facilities were also setup and tested (which was easier because there were so few students).

OB2: Twenty two minutes were spent covering the preliminary tutorial questions using the standard question response approach. However, as distinct from last semester students were primarily responding using audio rather than the text-chat. The teacher found it difficult to manifest a flowing conversation, partially due to the small number of students in the class. However, in response to short declarative questions, students gave similar sorts of responses (type and length) as if text-chat was being used (for instance, see Figure 108).

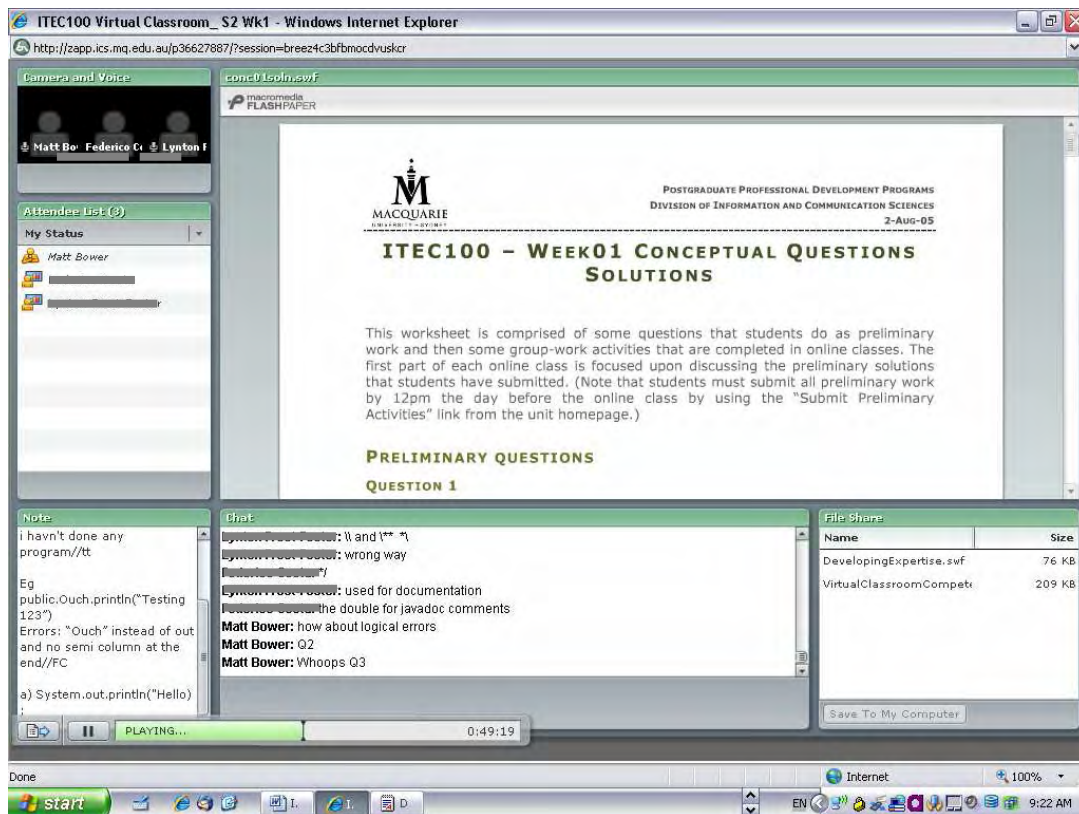


Figure 108 – Iteration 3 Topic 1 Covering tutorial questions using a similar approach to Iteration 1 and Iteration 2

The main difference was that the teacher provided a lot more elaboration in response to student questions. The audio appeared to support more conversational approaches by virtue of the ease with which students could contribute. The audio contributions provided by students in response to teacher prompts for solutions to Tutorial Question 3 are provided below. Note that although the duration of this conversation was longer than in Iteration 1 or Iteration 2, (6 minutes as opposed to 2 minutes and 5 minutes respectively) the depth of coverage by the teacher was greater. Also note that the examples provided by students were in textual form into the note-pod.

LF: I got this one wrong I think
FC: Syntax means it wont compile.
FC: E.g. public.Ouch.println("Testing 123")
Errors: "Ouch" instead of out and no semi column at the end//FC
LF: a) System.out.println("Hello");
FC: Are logic errors and runtime errors the same thing?
LF: Yeah, I think that's right.
FC: So the compiler doesn't eliminate all errors.
FC: I see. Thanks.

Note that for tutorial Question 7 when the teacher asked particular students to describe what happens behind the scenes when a computer runs a program, the explanations students provide using audio are far more elaborate than could have been accomplished using text-chat (in equal time). The audio contributions that students provide allow the teacher to assess that they have a relational (at least) understanding of how a computer runs a program. The pictorial model solution that the teacher subsequently presents is simpler than the explanations provided by the students.

OB3: Both LF and FC are capable students who ask some extension questions that are beyond the scope of the lesson. For instance, they enquire about the purpose of the parameters of the main method and the potential to override methods. This leads to a flowing discourse between the teacher and students. Students appear engaged by the capacity to adapt the direction of the lesson to meet their interests. A more conversational approach allows students to direct the lesson to meet their needs.

OB4: The teacher leads the debugging task involving the Cube program using the same (screen-sharing, prompting) approach used in the previous two semesters. However this time the students are using audio rather than text-chat to contribute their ideas. The teacher assumes a slightly more directive role than in previous semesters. The level and type of student contribution appear commensurate with that of previous Iterations.

OB5: The lesson finishes with a teacher-led discussion (with the teacher using screen-share to broadcast his desktop) regarding the interoperability (or lack thereof) of Microsoft word files and java files. This time a transmissive approach is adopted and correspondingly students make no contribution.

Key Incidents:

- KI1: Audio allowed students to provide faster and more elaborate contributions than was possible with text-chat, though only one person could contribute at a time. Audio appears to represent an appropriate modality for representing rapid textual contribution by a particular student.

- KI2: In the teacher-led class debugging task the amount of teacher discourse and teacher prompts applied determines the level of student discourse, which is similar to previous semesters even though students are using audio.
- KI3: One student (TI) leaves the class half way through and does not return. The level of difficulty appears to intimidate this student.

Reflective Notes:

- RN1: The use of student audio appears to offer advantages over text-chat in some situations but not others. For instance, because only one person could make audio-based comments at a time the rate of contributions per person may decrease as the number of group members increases.

Iteration 3 Topic 2

Topic: Fundamentals of Objects and Classes as they relate to object oriented programming.

Attendees: 3

Summary of Lesson

OB1: Preliminary Tutorial question 1 was completed using the same interface as in Iteration 2, however the task had been adjusted to account for the incorrect program code used in the previous semester (see Figure 109). Correspondingly the need for inter-student discourse relating to the discrepancy between the program code on the worksheet and in the group-work room was obviated, and students progressed with the task more confidently.

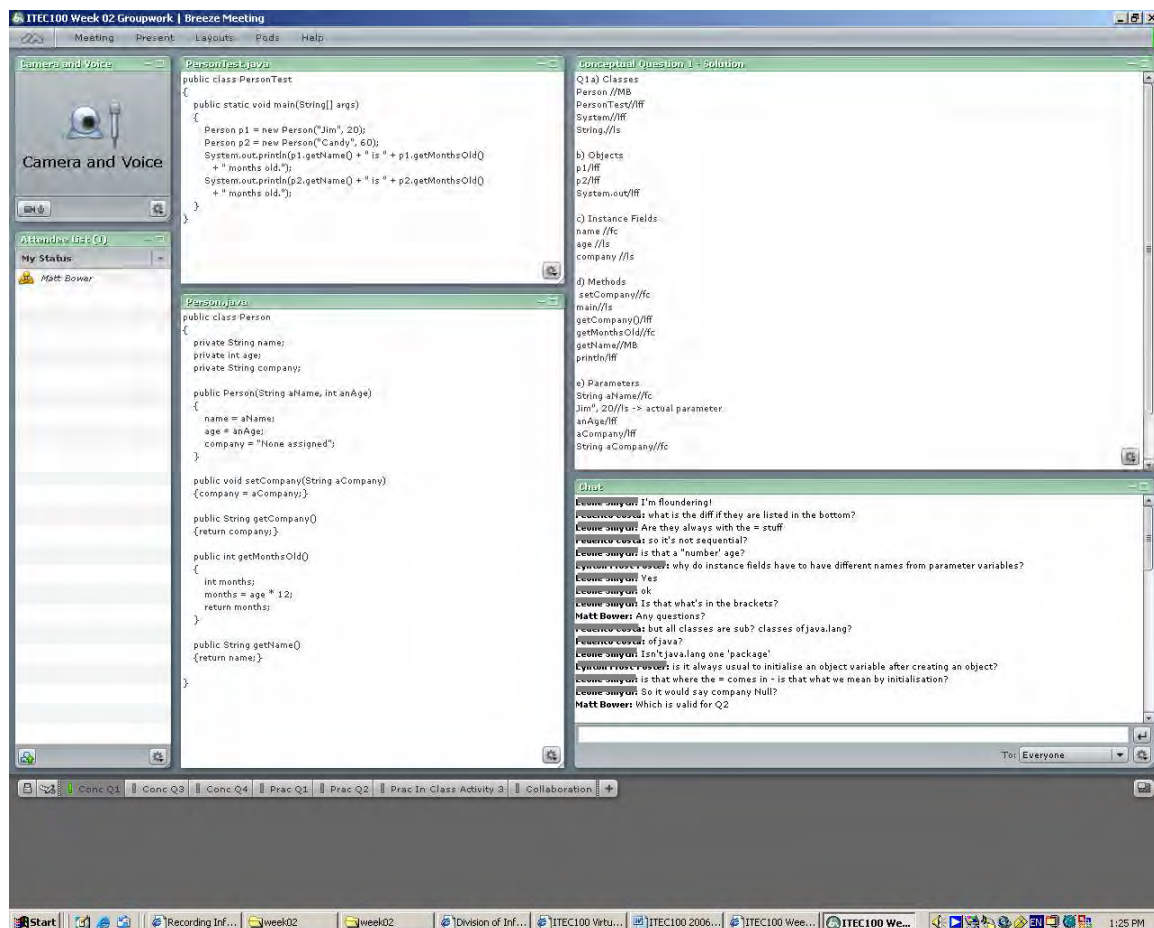


Figure 109 – Iteration 3 Topic 2 Student-centred sharing of declarative knowledge using audio

The teacher and students held considerable discussion during this task. Students also asked spontaneous and relevant questions after the task that covered many conceptual matters required for understanding this week's work, as well as future week's content. This was an example of adapting the task in light of student descriptions, as part of Laurillard's (2002) Conversational Framework. For instance, student questioning led to a discussion about `System.out.println` being a method of the `System` class, and that all classes were subclasses of

a universal “Object” class. Because of the elaborations being discussed, coverage of question 2 did not start until 30 minutes after question 1 had been commenced.

OB2: The teacher covered question 2 through to question 5 in the main classroom using a teacher dominated question-response approach (to speed up collaborations). At times students ask questions using text-chat, but at other times by using audio. For the question 3 “true or false questions” the teacher asks students to use text-chat as an approach to gauging all of their conceptions simultaneously. Question 5 is finished 58 minutes after the tutorial question review commenced. The teacher’s more directive approach successfully increased the pace with which these final four tutorial questions were covered to compensate for the time spent on question 1.

OB3: After the break the practical questions are discussed using the same approaches as semester 2, except students have audio facilities to make their contributions. Practical task one comparing the two approaches to raising the salary resulted in well reasoned evaluative student discussion. When students are asked to download and unzip practical task 2 there is a long period of silence while students download and open the files using their IDE. Having the learning artefact in each individual’s private space (on their machine) prevents a common focus for activity and discussion from being easily identified. This effect provides confirmation of the effect that was observed in Iteration 2. The teacher then dominates conversation when covering the practical tasks, again to account for lost time.

OB4: Because of the extended time taken to complete the preliminary activities, the teacher refers students to the downloadable solutions for the last exercise.

Key Incidents:

KI1: One student struggles to establish audio communication at the beginning of the lesson.

KI2: Students use audio and text-chat at different times for different purposes. For instance, using text-chat when the teacher was talking allowed students to ask a question without interrupting.

KI3: As in Iteration 2, asking students to download and unzip files to compare and contrast them is less efficient for small programs than having the programs available to inspect side by side in the note-pod. Placing the files in the note-pods also has the advantage of providing the resources in an accessible space that can be edited by anyone for explanatory or hypothetical purposes.

KI4: As was the case in the previous week, students request that the size of the screen-share be enlarged and the teacher once again explains that students should use the scroll button. (Note that in Iteration 1 this response was not automatically provided by the teacher, meaning the improved web-conferencing competencies of the teacher also allows student to improve whole class collaborations.)

Iteration 3 Topic 3

Topic: Introduction to types and numbers

Attendees: 3

Summary

OB1: The preliminary conceptual work for this week involved several declarative knowledge questions. The students were asked to derive answers to the first seven questions as a group in the main classroom on a note-pod (see Figure 110).

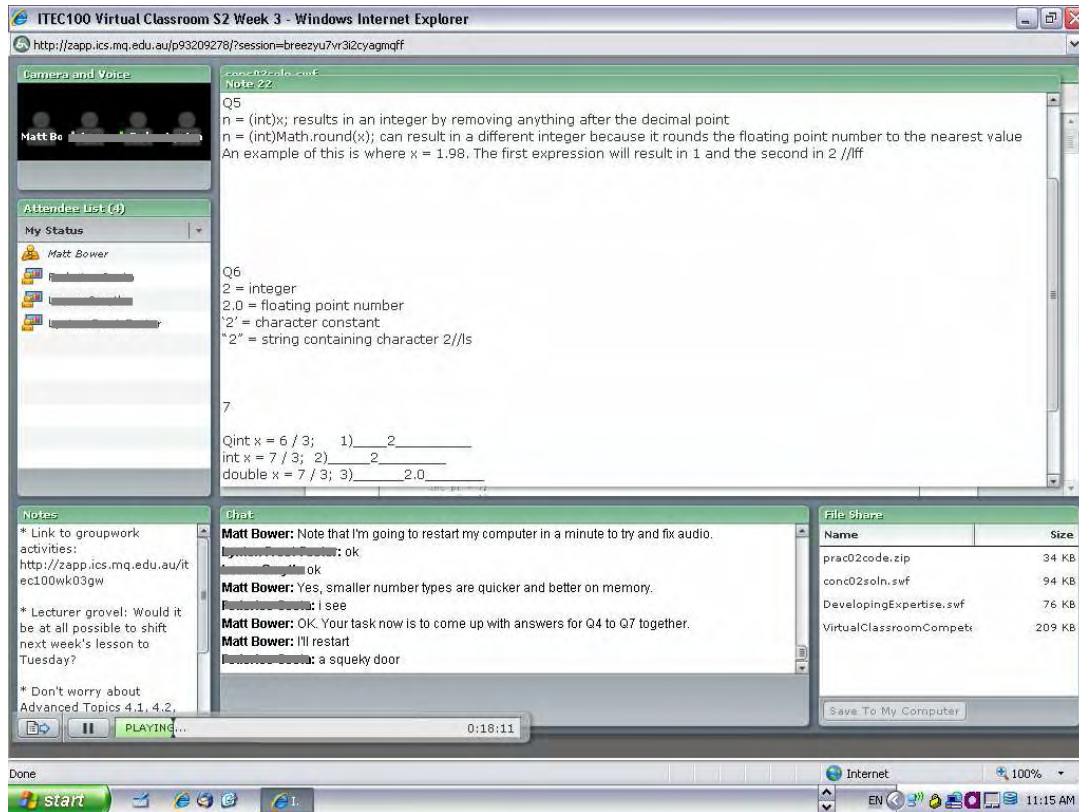


Figure 110 – Iteration 3 Topic 3 using a note-pod for composing group solutions to tutorial questions

The teacher and students then discussed the solutions. A conversational approach is used to cover questions 8 to 10, with the students asking questions regarding content areas about which they were unsure (such as variable name choice and use of the Math.pow function).

OB2: The teacher covers the last three questions (11 to 13) using a teacher-centred (transmissive) approach to save time and so pre-prepared diagrams can be utilized. Visual representation allows the modality to more appropriately represent the information that is being communicated, in accordance with Salomon's (1994) Symbol System Theory. During this time students ask two questions using text-chat rather than audio:

FC: so math is a static class?

LS: are they all standard maths functions?

The teacher is then able to respond to these questions by screen-sharing the java API specification and answering by showing specific examples.

OB3: After the break the teacher displays a student's SquareRoot program in a note-pod and discusses it. The program is then pasted into the teacher's IDE and shown running on the teacher's computer using the screen-share facility. This then provides a pattern of collaboration for the first group programming exercises (requires students to write a temperature conversion application), allowing them to rapidly commence the task without verbose explanation.

OB4: The students write the Temperature program in the group-work room which had been purpose designed for the task (see Figure 111 for a screenshot of the room design). A large note-pod is used for students to collaboratively author the program. Based on teacher encouragement, they adopt the convention of placing their initials after a line of code that they write (to identify the contributor to others in the group, facilitating discussion). Students all use audio to collaborate which allows for dual processing (Low & Sweller, 2005). Using audio in combination with the note-pod also reduces the level of split attention (Ayres & Sweller, 2005) as students do not have to monitor two textual input pods simultaneously (the note-pod solution space and the chat-pod). Focus is promoted by having the solution space large enough to present the entire program code at once (avoiding the need for scrolling).

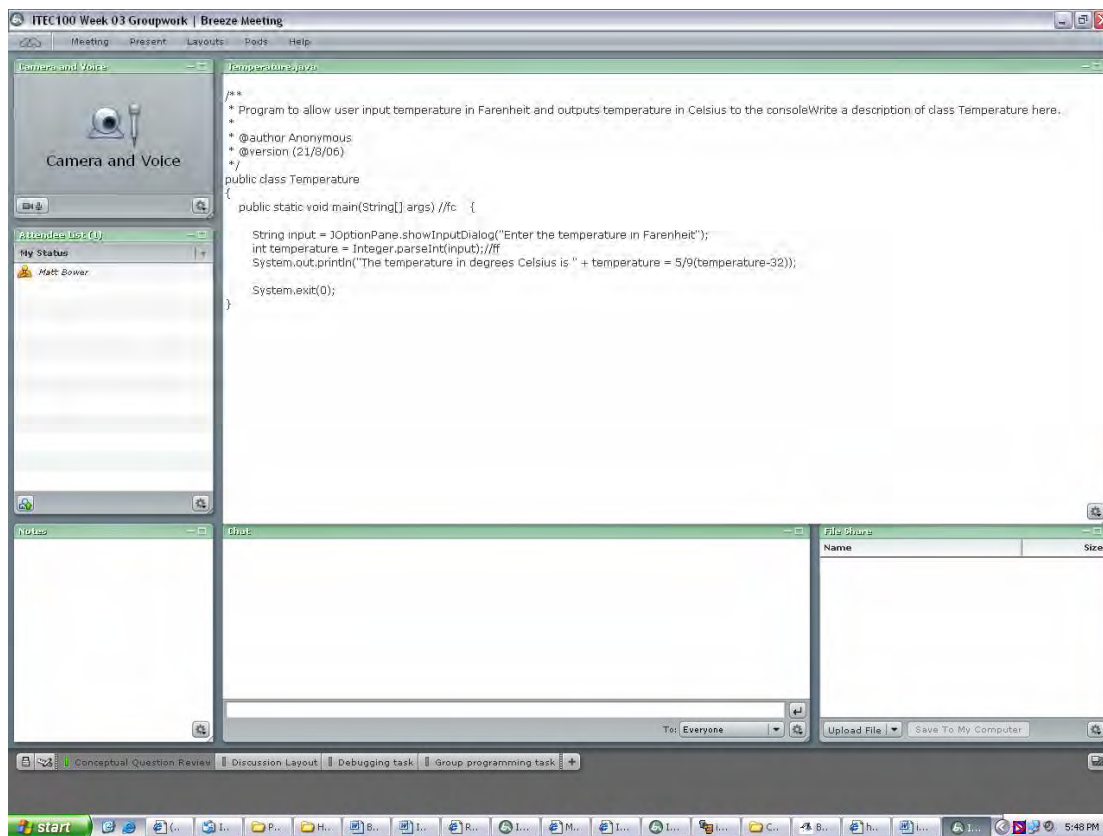


Figure 111 – Iteration 3 Topic 3 The group-work room used for group programming of the Temperature program

They run the code on their own machine by copying and pasting from the note-pod to their IDE. At times the fact that students are working on the problem on their own machine lowers the rate of collaboration and leads to more individual rather than shared cognition. The small note-pod in the bottom left corner of the interface, the chat-pod in the bottom centre, and the file-share pod in the bottom right of the window are not used. Students adequately perform the task using the audio (still being broadcast from the main room) and the one large note-pod in the top right corner of the window. The interface would have removed these unused pods (redundancy principle, Sweller, 2005b). Due to time restrictions the teacher discusses (and demonstrates using screen-sharing) how to debug the final two errors in their group program.

OB5: The teacher uses a standard sharing interface to broadcasts a students' Square program in which methods returning the diagonal and area of a square were required. The teacher uses the program to hold a discussion regarding program design (for instance, whether the length and diagonal should be instance fields if the program is to return the diagonal). Throughout the teacher-led discussion students contribute well considered thoughts to the specific issues raised. Improvements to the program are made by the teacher on the basis of the discussions.

OB6: The final activity requires students to write a TinCan class. The teacher broadcasts his screen using a standard sharing interface displaying the IDE. Students are asked to instruct him on how to change the previous square program so that it returns the surface-area and volume of the cylinder. During this teacher-led episode the teacher often prompts students for input and raises several issues relating to aspects of programming. For instance, constructing objects with multiple attributes, matching object attributes to variables, mathematical functions, code reuse and design patterns are all discussed. As the TinCan activity continues the teacher progressively adopts a more directive and dominant approach rather than a conversational one (due to time restrictions). This results in more tactical rather than strategic questions from the teacher, relating to individual lines of code rather than stimulating discussion about broader and more strategic design issues. For instance, the teacher intentionally leaves small errors in the program for students to debug (which they successfully correct). This means that the teacher can only assess a multistructural level of student understanding as students do not have the opportunity to demonstrate that they can strategically interrelate all aspects of the knowledge required to meet the program specification.

Key Incidents:

- KI1: As audio becomes pervasively adopted by students text-chat is seldom used.
- KI2: Due to a problem on his machine the teacher lost audio capabilities during the beginning of the lesson so used text-chat during preliminary tutorial questions 1 to 3.
- KI3: Students use text-chat to ask questions while the teacher is adopting a transmissive approach to instruction. Even though students have audio setup the text-chat medium allows them to contribute without interrupting the teacher's broadcast.
- KI4: The interface design and activity design for writing the temperature-converter program overcomes several problems observed in previous lessons. It should be noted that using the note-pod approach to group programming can lead to more individual programming efforts on people's and less shared cognition than if screen-sharing is being used. On the other hand screen-sharing can lead to one student dominating the programming effort. One or other of these approaches may or may

not be appropriate depending on the teacher's objectives for the learning activity. The teacher can influence the level of shared cognition by prescribing tasks where activity remains in the collaborative space.

- KI5: During the square example appropriately pitched and stimulating questions are perceived by the teacher to result in valuable discussion.
- KI6: The TinCan programming exercises is appropriately pitched for the class and as such several pertinent issues arise in discussions.
- KI7: During the TinCan example the students and teacher spend some time experimenting with the full-screen toggle feature, which when setup correctly allows students to elect to see the teacher's desktop across their entire screen and still make comments using audio. This is the optimal approach to conducting screen-sharing activities.

Reflective Notes:

- RN1: Intentionally focusing on more tactical issues during the TinCan class programming activity may be more appropriate at these early stages of the subject whereas encouraging more strategic discussions may be more appropriate once students have mastered the syntax and semantics of writing programs.

Iteration 3 Topic 4

Topic: Introduction to Applets and Graphics

Attendees: 3

Summary of Lesson

OB1: The same activity and interface design for preliminary tutorial questions 1 to 3 was used as Iteration 2, except all students had audio capabilities (see Figure 112). In terms of task prescription the teacher designated each of the three students responsibility for starting one of the three questions and requested that they then embellish the answers provided in the other note-pods. This ‘division-of-labour’ (Engeström, 1987) by the teacher effectively avoided the need for students to hold activity coordination discussions relating to how complete the task.

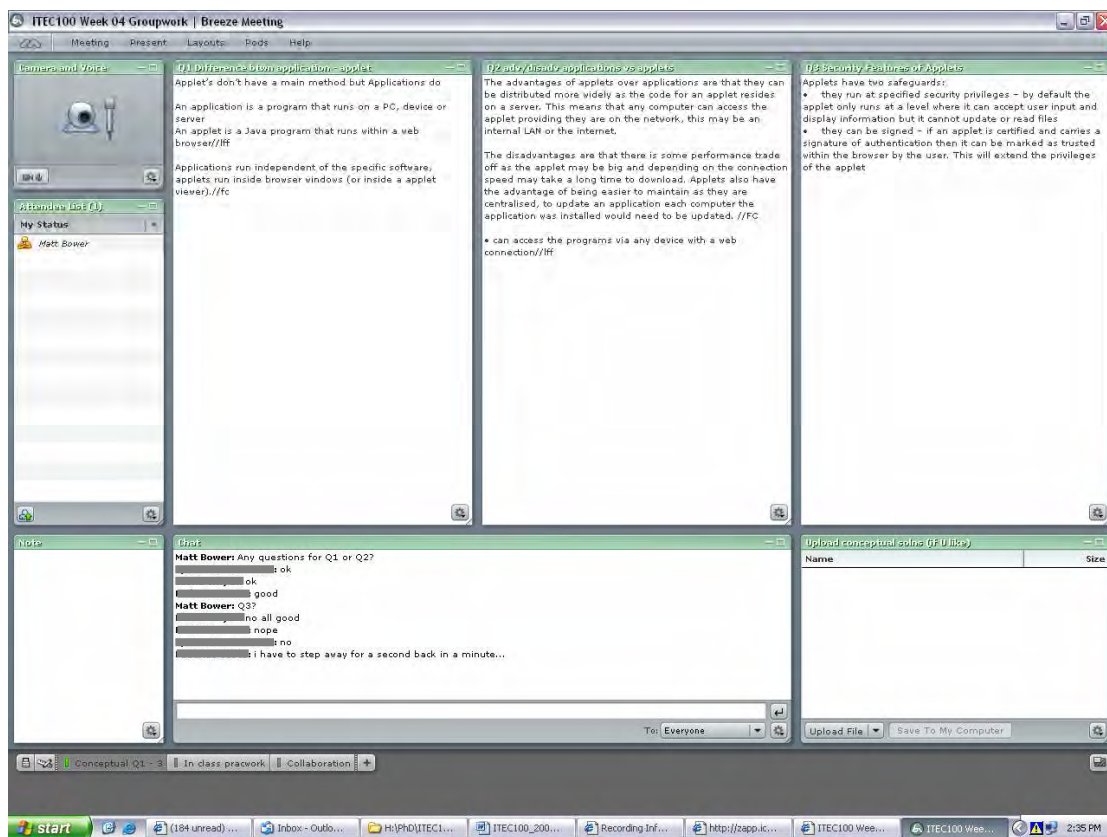


Figure 112 – Iteration 3 Topic 4 Retest of the applet comprehension question using audio

Students spontaneously added initials to the end of their note-pod contribution, indicating that the collaborative pattern was becoming a standard behaviour, and obviated the need for students to discuss who had contributed each piece of content. Students then added points to the contribution made in other pods. Note that limited discussion regarding the subject matter resulted from this exercise.

OB2: A teacher-led question-response approach is adopted for preliminary tutorial questions 4 to 6, with some periods of teacher-centred transmissive approaches. During one of the teacher's instructive sequences a student asks the following question using text-chat:

FC: But you always have to call the graphics first can't call the graphics2d first?

Once again the use of text-chat allows the students to avoid interrupting the teacher's commentary, and the teacher is able to respond once he had finished the point he is discussing. The six tutorial questions are finished within 16 minutes allowing the class to proceed directly to the practical activities.

OB3: The first practical task requires students to debug and improve a student's applet drawing of a flower. A purpose built interface has been designed to facilitate this, with the entire program code visible in the two note-pods (see Figure 113). This was an attempt to create more negotiated and collaborative approaches to meaning making than had arisen from the applet drawing task in Iteration 1 and Iteration 2. Students were instructed to load the program up onto their machines from the zip file they had previously downloaded so they can work on it on their own machine. There was silence for almost three minutes while the students (and teacher) loaded the project into their IDE.

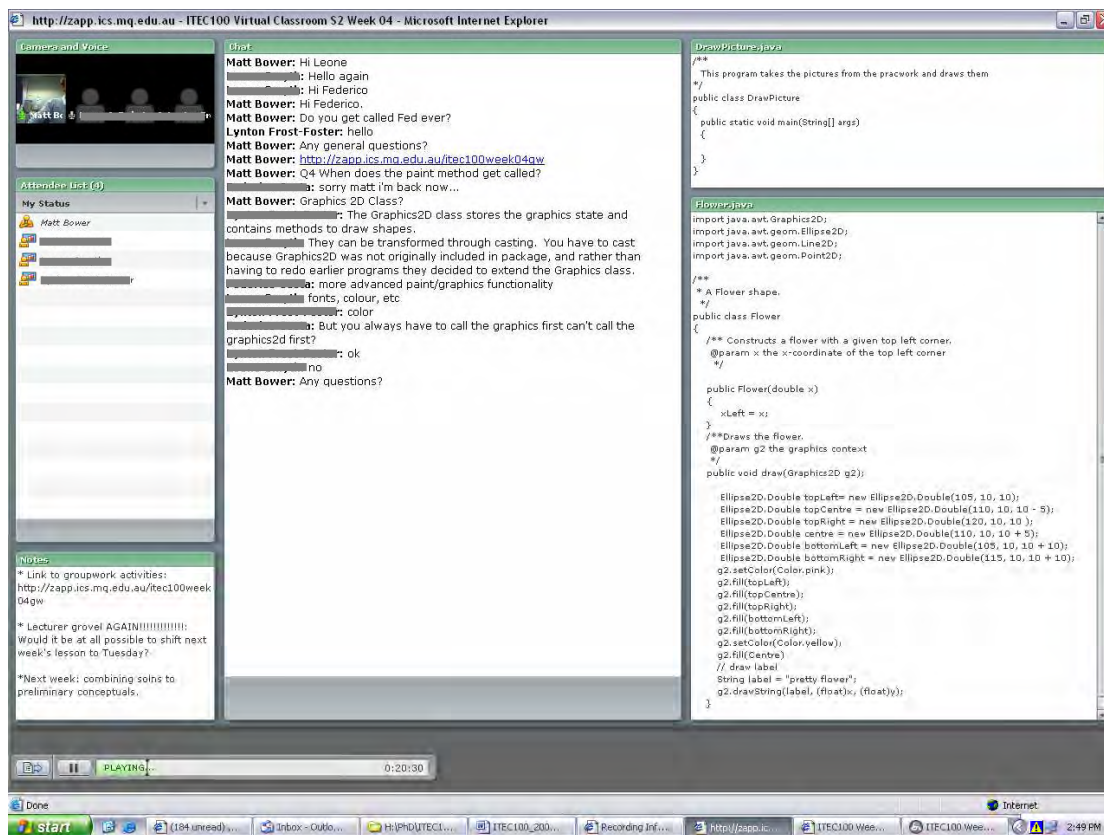


Figure 113 – Iteration 3 Topic 4 Original interface for the debug flower applet task

The teacher explains that the program is to be changed so that it accepts the coordinates of the top left corner of the flower and allows it to be drawn at a position determined by the user. The teacher makes the first adjustment to the constructor by editing the note-pod and

leaving his initials. This sets a model for the way students are expected to collaborate. Students make little progress on the task initially, and ask questions regarding constructing different drawing objects in different classes, extending the Applet class, and the Graphics class. The teacher responds to these questions. Students are struggling to contribute and the teacher encourages them to ask more questions if they require assistance to proceed. One student (LF) asks about the coordinate system (using audio):

LF: Does that mean that it starts at the top left corner of the screen?

The student has difficulty understanding how providing the x and y coordinates of the flower relates to the coordinate system of the canvas. The teacher spontaneously adjusts the interface to incorporate a whiteboard, allowing him to represent the relationship between the flower and canvas coordinate systems. This is in accordance with Salomon's (1994) Symbol System Theory which advocates representing information in the most cognitively efficient form, and with the multimedia learning principle (Fletcher & Tobias, 2005). As well as supporting the teacher's explanation, the whiteboard allows students to represent their amended conceptions so that the teacher can gauge that they have accurately developed mental models (see Figure 114).

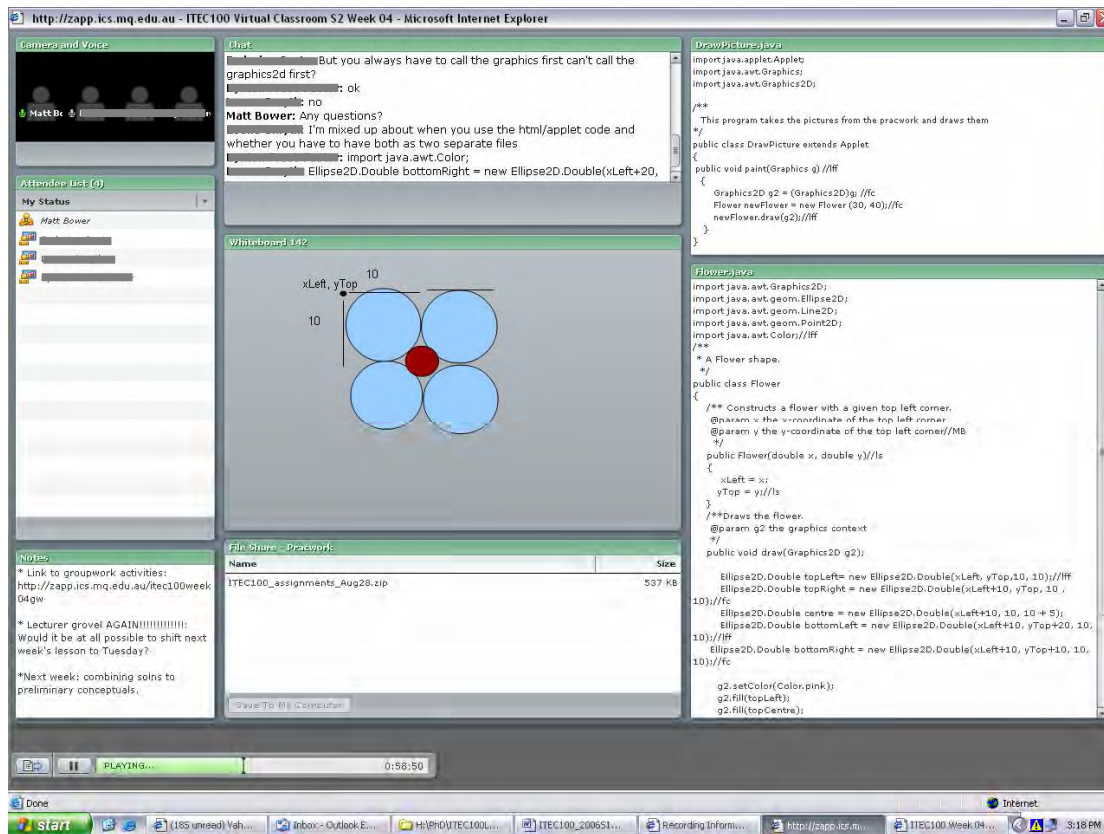


Figure 114 – Iteration 3 Topic 4 Interface adjusted to support conversation during the debug applet flower task

Students then adjust the flower program to incorporate the provision of x and y coordinates in the constructor. Other questions regarding aspects of programming such as the relationship between applets and complex drawing objects are asked and discussed, broadening out the curriculum matter addressed beyond the original scope of the task. Sixty

minutes was spent on this task by the time the initial objective has been completed. Based on the teacher's observations and student comments, the questions asked and the discussions held were all relevant and effectively supported student learning, even though it was sometimes not directly related to the task. Teacher adaptation of the goals of the learning episode based on student feedback allowed the direction of the learning task to be adjusted to better meet student needs. This is a flexibility afforded by applying the feedback cycle of Laurillard's (2002) Conversational Framework. As an extension to this, the interface was then able to be adjusted accordingly, in order to meet the changing cognitive and collaborative requirements of the learning episode.

OB4: The teacher then asks how they might utilize the new flexibility of their program to draw several flowers in random positions. The teacher then spends six minutes accomplishing this task using a predominantly transmission approach in order to cover the task in a time efficient manner.

OB5: Due to time restrictions caused by the extended conversation during the flower applet activity, the teacher also uses a presentational approach to demonstrate how other objects can be incorporated into the applet drawing, and how documentation can be automatically generated from the java files to support this object oriented approach to programming. The standard sharing interface is adopted (see Figure 115). Note that students still use text-chat occasionally, but usually only when the teacher is adopting a transmissive approach.

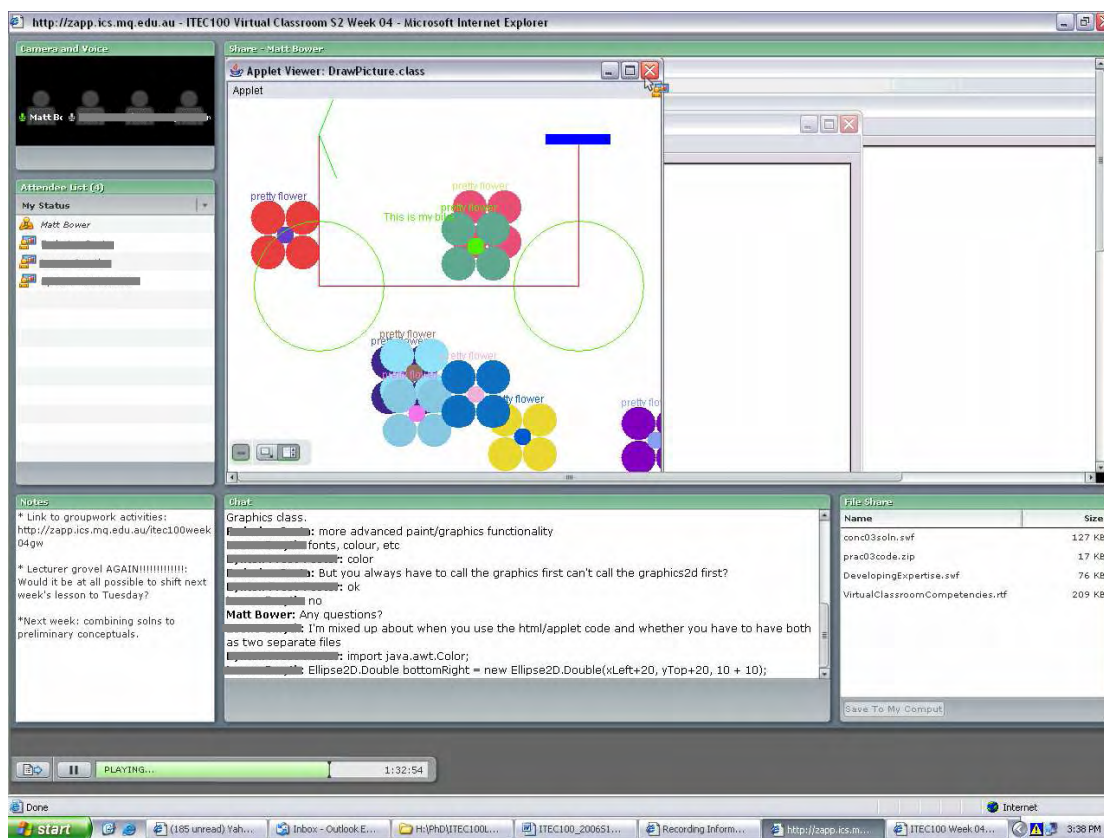


Figure 115 – Iteration 3 Topic 4 Transmissive screen-sharing approach to demonstrating program

OB6: After a short break a final task is presented to students requiring them to draw a time on a clock face based on user input. The teacher asks students to instruct him on how to accomplish this while he broadcasts his IDE using a standard sharing interface. The teacher-led task takes 24 minutes to complete and results in a wide variety of discussions regarding the Arc class, trigonometry and transformation of user input (all relevant to the first assignment).

Key Incidents:

- KI1: The task prescription incorporating role specifications for the group-work activity in preliminary tutorial questions 1, 2 and 3 allowed students to focus upon the conceptual material rather than coordinating interactions.
- KI2: The first practical task requiring students to debug and augment the flower applet resulted in conceptual questions that did not arise during the tutorial questions.
- KI3: In order to support the understanding of a multifaceted, spatial concept during the Applet programming task the teacher spontaneously elects to use a whiteboard. Accompanying audio media with a spatial representation utilizes the multimedia principle (Fletcher & Tobias, 2005) and modality principle (Low & Sweller, 2005), resulting in a clearer explanation.
- KI4: When students are in full-screen mode they can only contribute audio if they have their audio button locked to the “on” position. However for some students this causes the whole room to echo so they tend to leave the audio button in the off position. This means that they cannot make audio contributions while viewing screen-sharing in full-screen mode. This is only determined during the lesson.

Reflective Notes:

- RN1: The fact that the flower applet activity took an extend period of time did not imply the learning task was inefficient. Tasks should be seen as catalysts that spark relevant and effective learning conversations. A task that is conducted directly and superficially by one group of students may be extracted yet enriching in another class based on the depth and breadth of content covered (the breadth often reaching beyond the learning objectives anticipated for the task).
- RN2: The teacher supported practical tasks were effective in promoting discussions in line with the Conversational Framework (Laurillard, 2002).

Iteration 3 Topic 5

Topic: Introduction to conditional statements ('if' statements)

Attendees: 3

Summary

OB1: In order to promote more collaboration outside class in a fashion similar to Topic 5 of Iteration 2, students were set the task of completing the preliminary conceptual activities in a group using the web-conferencing system. The repeated trial of this approach during both iterations could then allow similarities to be gauged and contrasts to be drawn. The only difference in the approach adopted for the pre-class group-work this semester was the pervasive use of audio (rather than text-chat) to hold conversations. This semester students spent considerably less time completing the Topic 5 preliminary tutorial questions than previous Iteration (1.5 hours as opposed to up to 4 hours the previous semester), although the answers provided were less elaborate.

OB2: The first part of the lesson was spent discussing the combined student solutions to the tutorial questions. The capacity to inspect the students' group solutions before class allowed the teacher to efficiently identify areas of conceptual weakness amongst the group and hence provide accurately targeted instruction. For instance, students had not been able to derive an answer to question 8 regarding side-effects. As a consequence the teacher spent some time discussing this using audio and the model solutions broadcast using the standard sharing interface.

OB3: The students were once again asked to represent the output of the if-then-else statements of Question 3 on the whiteboard. However this time the question was placed next to the corresponding number line in order to reduce split attention between the tutorial sheet and the solution space (see Figure 116). As well, students used audio rather than text-chat to collaborate. Students were initially allocated to one task each. This initial allocation of one student per part alleviated the need for them to discuss who would be performing which role. Using audio enabled discourse to be contributed and interpreted at the same time as people's solutions (leveraging dual processing capabilities, Low & Sweller, 2005) as opposed to text-chat which represented another visual channel that could result in split attention (Ayres & Sweller, 2005).

OB4: As was the case in the previous semester students were asked to provide formal feedback through the polling tool and note-pod regarding their impressions of working collaboratively on the preliminary tutorial questions as opposed to individually. All three students responded that it was better to complete the questions collaboratively. Advantages raised included that it provided support with the more difficult questions and that students could participate without having a complete understanding. Disadvantages suggested were that it was hard to find a suitable time each week to meet and that it was another night that they were committed to work.

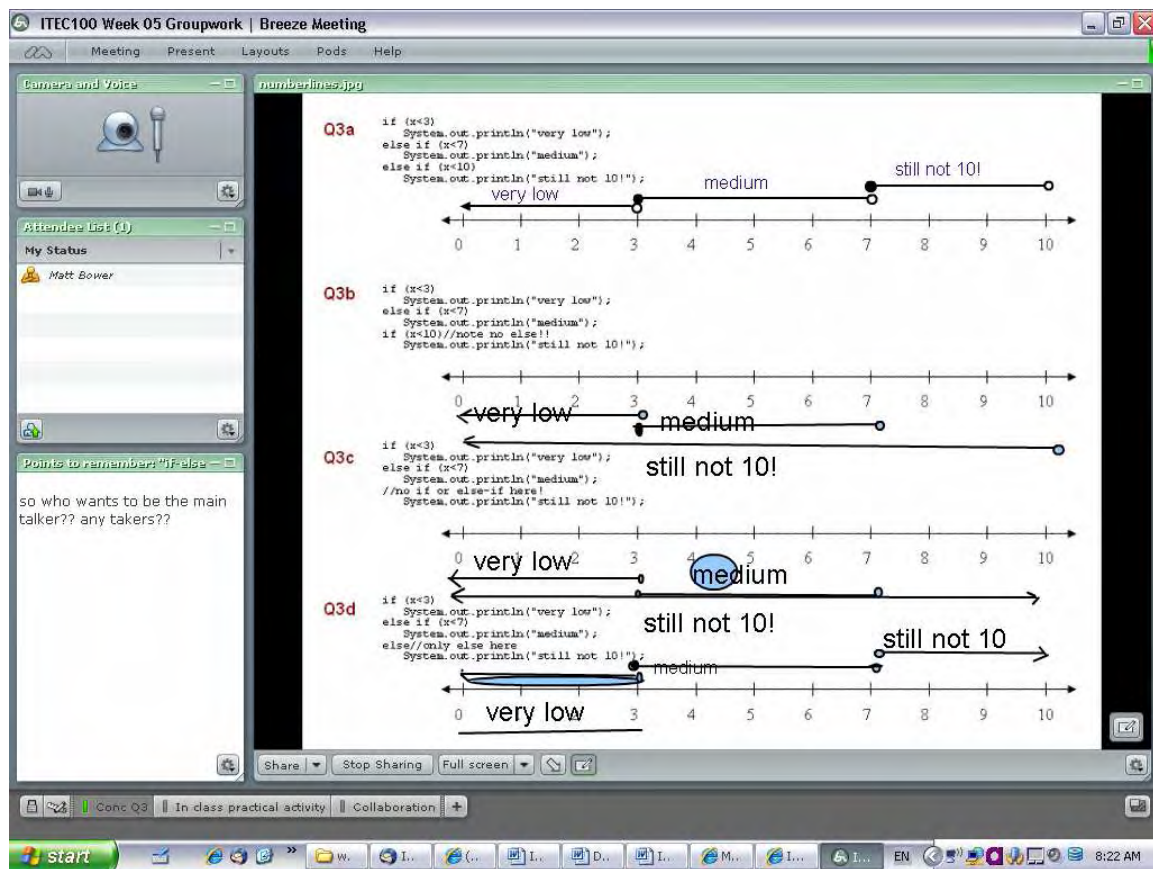


Figure 116 – Iteration 3 Topic 5 Augmented student-centred approach to sharing conceptual information

OB5: The practical work was covered using a similar approach to the previous week, although at times the teacher adopted a more dominant role in the conversation. Students agreed to share and review each others' code, and suggestions about how to improve the programs were discussed. The teacher broadcast his screen with students' Month, Tax, InCircleApplet and InRectangleApplet programs being displayed in the IDE while they were being debugged and considered by the class. Debugging the students' programs led to several questions regarding syntax and how to perform programming processes to be posed and resolved. Comparing and contrasting different students' solutions broadened students' understanding of the range of problem solving approaches available to them. The discursive approach allowed the level of students' mental models to be gauged and instruction adjusted accordingly.

OB6: The group programming activity required students to merge the Month and LeapYear classes that they had written so that the user could input the year and month and then the correct number of days in the month were returned (including February of a leap year). This task was conducted in a purpose built interface with note-pods containing the relevant code as a starting point (see Figure 117).

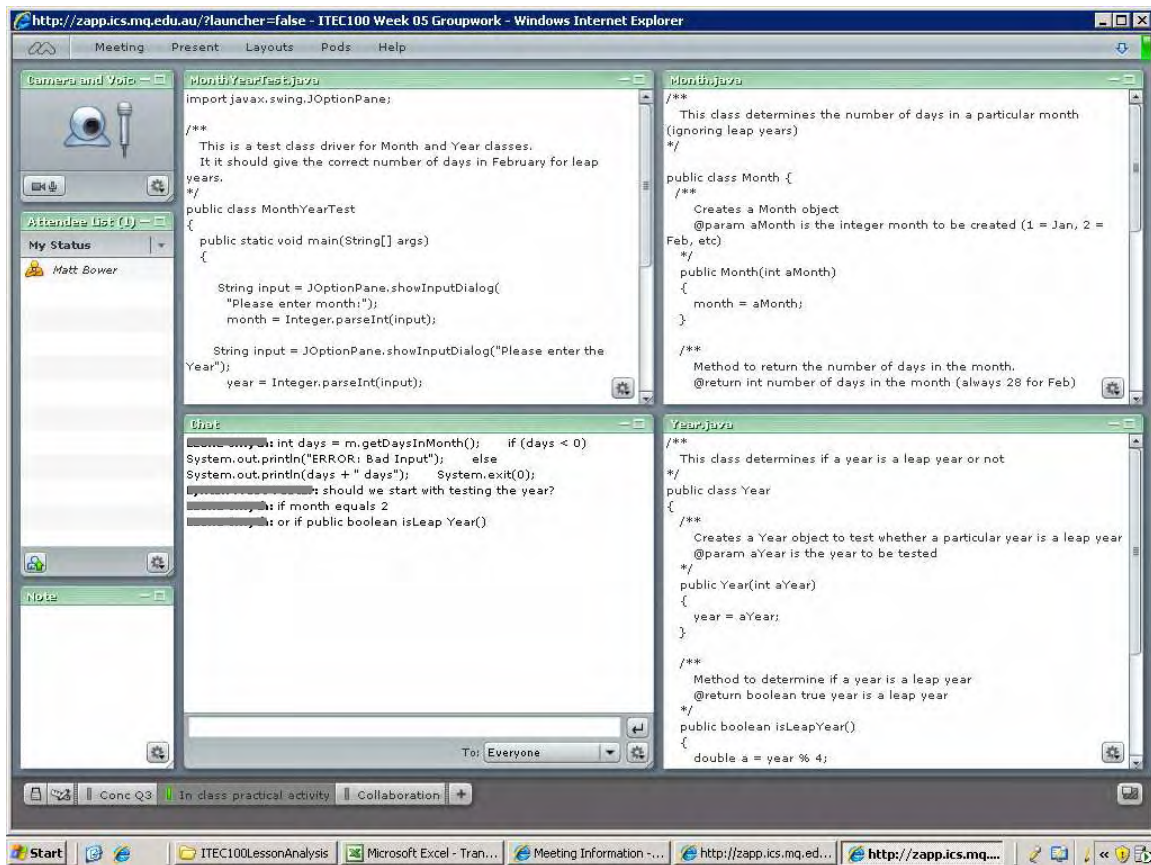


Figure 117 – Iteration 3 Topic 5 Multiple note-pods approach to group programming

The interface allowed all code required to be viewed in one window and provided equal access to all students. This is in contrast to having a screen-sharing approach where the code is segmented between several files and one person has control of the editing process. As part of the student-centred activity design the teacher intentionally withdrew himself from the discourse so that students could play a more active role in the problem solving process. This resulted in higher levels of student involvement than in teacher-led programming tasks. This not only allowed students to negotiate meaning amongst themselves but also allowed the level of students' understanding to be more accurately gauged. Students took approximately 17 minutes to complete the exercise.

Key Incidents:

- KI1: Group-work approaches to pre-class activities allow the teacher to more efficiently identify the conceptual weaknesses of the "group" as opposed to specific individuals. This is appropriate if remedial instruction is to the "group". In this way the teacher is addressing the "combined" cognition of the group.
- KI2: For the task requiring students to combine the month and leap-year programs, arrangement of the note-pods to show all three project files allowed different students to be working on different files at the same time. This also allowed all students to relate code between files. The shared artifacts of activity (objects) were combined in the one interface and students had an authentic collaborative task to perform, in accordance with recommendations for successful online collaboration made by Jonassen (2000). The task allowed student mental models to be revealed.

Iteration 3 Topic 6

Topic: Introduction to iteration ('loops')

Attendees: 2

Summary

OB1: Student solutions to the tutorial questions were broadcast in the main room using the standard sharing layout. These were discussed by the teacher and students. The teacher demonstrated how to use the debugger to find the errors in the preliminary tutorial question 2 factorial program, which also provided a dynamic representation to support the development of students' "notional machine" (du Boulay, O'Shea, & Monk, 1989) regarding loops. Broadcasting the IDE allowed process information relating to debugging programs to be shared (see Figure 118).

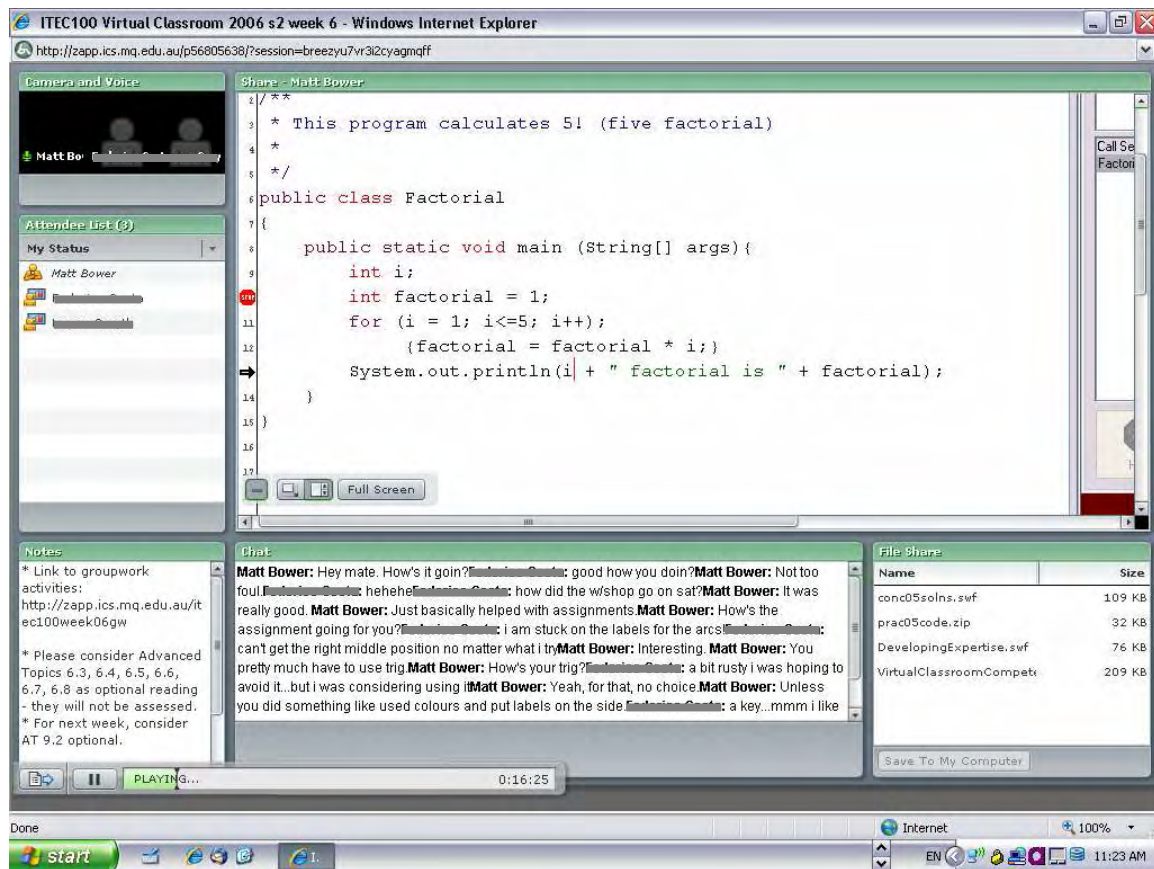


Figure 118 – Iteration 3 Topic 6 Teacher screen-sharing debugger to demonstrate how to solve erroneous Factorial program

For preliminary tutorial question 6 regarding scope the model solutions were presented. This allowed a concrete and accurate answer to an elaborate question to be provided in a short span of time.

OB2: Following the break the same two in-class conceptual 'loop' activities as in Iteration 2 Topic 6 were conducted. The same interface design was adopted, except students used

audio. In terms of task prescription, the teacher explicitly requested that students used logic to solve the problems rather than use the IDE to run the programs. The conversation that resulted was excellent at exposing students' mental models. Audio allowed rapid exchange of thoughts between the students, as well as enabling them to easily contribute to the solution space at the same time as they discussed ideas and negotiated understandings. The teacher deliberately removed himself from collaborations for long periods of time, which allowed the students space to contribute. When the teacher chose to re-enter into the episode it was possible to offer accurately targeted remedial instruction and appropriately pitched questions, based on the understandings that students had revealed through their discussion. This is an example of the "feedback" stage of Laurillard's (2002) Conversational Framework based on students' "apprehension", "interpretation" and "application" of content.

OB3: As in Iteration 2, after students had attempted the activity the teacher ran the second (nested loop) program in the debugger while broadcasting his screen. The audio commentary and screen-share provided a lesson both in nested loops and how to use the debugger. The way in which the debugger showed how the program was executed line by line, how variables were created and incremented, and how calculations were output again provided students' with a dynamic model that could support the development of their notional machine (du Boulay, O'Shea, & Monk, 1989).

OB4: The teacher then reviewed the practical work using a standard sharing interface displaying the IDE. Student attempts at the CoinFlipper, NameReverser and RandomCircle programs were shown and discussed. Although students contributed some comments and questions throughout, the teacher adopted a principally presentational approach which dominated the discourse.

OB5: The group programming task required students to create an applet that drew a chess board. Due to time restrictions the teacher decided to lead this task, broadcasting his screen and fielding instructions from students about how to complete the task. Because it was not originally anticipated that the teacher would lead this task, setting up the starting point took almost five minutes. After a few minutes of using screen-sharing the teacher also decides to integrate a note-pod into the interface so that students can be performing the programming. However, the note-pod is small and only displays a small section of the source code (see Figure 119). There are two unused pods on the lower right hand side of the window that should have been removed in accordance with the redundancy principle (Sweller, 2005b). This was an example of the teacher attempting to dynamically adjust the interface to engage distributed activity but not creating a design that effectively supported the activities required. In hindsight it is apparent that the teacher needed to decide between a screen-sharing (teacher-led) or note-pod (student-centred) approach. Alternately one approach could have been followed by the other. However incorporating both forms of textual representation at the same time caused split attention (Ayres & Sweller, 2005). At one point the teacher enlarges the solution note-pod so that it overlaps the screen-share in order to allow more of the program to be displayed at once. While this increases the focus on the solution space it interferes with the screen broadcast. Despite the sub-optimal interface the teacher and students hold an extended (23 minute) conversation to collectively derive the solution.

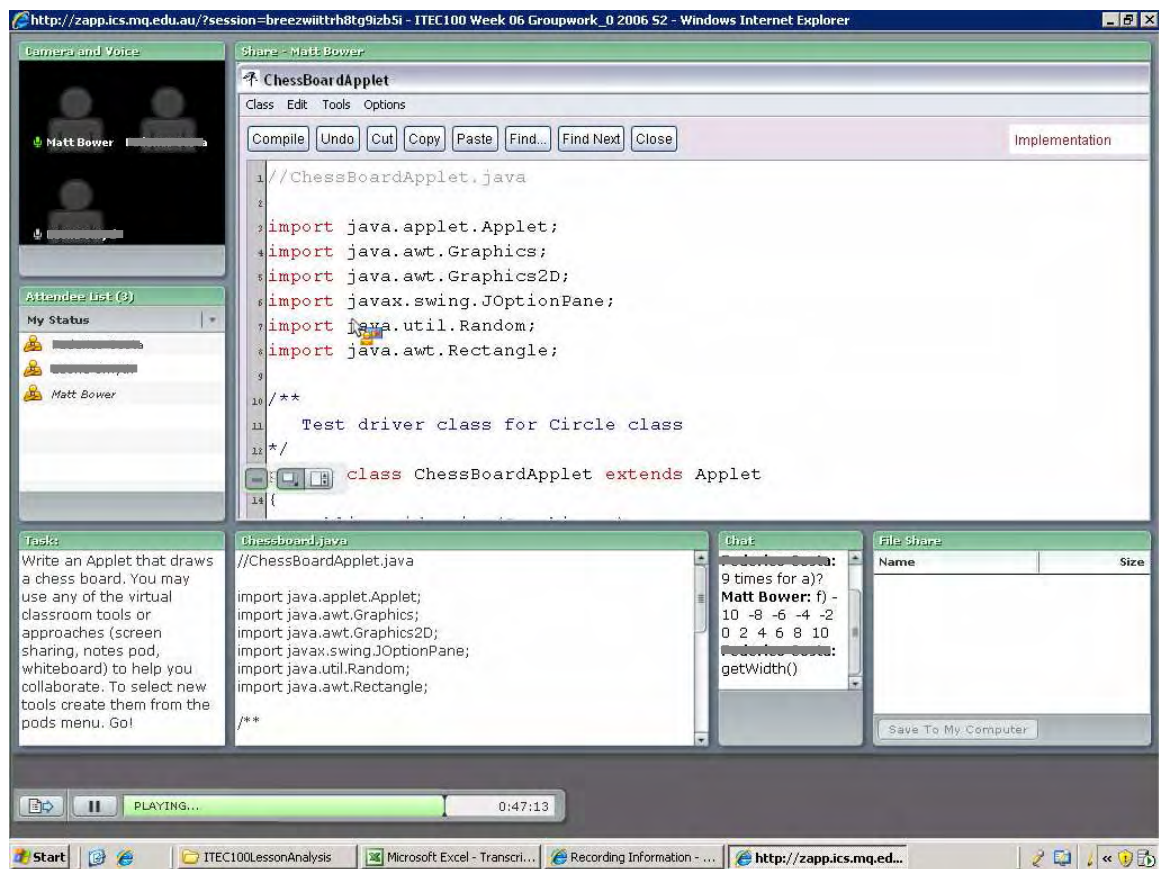


Figure 119 – Iteration 3 Topic 6 Attempt to spontaneously adjust interface to meet collaborative and cognitive requirements of episode

Key Incidents:

- KI1: The use of the debugger on occasions throughout the lesson provided an exceptionally clear approach with which to support development of students' notional machine (du Boulay, O'Shea, & Monk, 1989).
- KI2: While there were instances where the teacher applied full screen-share mode so students could have the optimal interface for viewing the IDE, there is also an instance where he started discussing a program while unintentionally not turning screen-sharing mode on.

Reflective Notes:

- RN1: The successful collaborations that were observed during the two in-class 'loop' activities coincided with:
 - an appropriately pitched task
 - a well designed interface (facilitating rapid communication through audio, student solution spaces with no split attention present)
 - a task specification clearly identifying the type of collaboration required, and
 - the deliberate withdrawal of the teacher from the students' collaborative space.
- RN2: With less students in the class, conversational approaches may be easier to apply (because students feel less intimidated in sharing their thoughts, and are less able to hide behind the contributions of other students).

Iteration 3 Topic 7

Topic: Introduction to polymorphism.

Attendees: 2

Summary

OB1: The teacher adopts the same approach to opening the lesson as the previous semester, which involves broadcasting the whiteboard containing the four files of the Polymorphism program (see Figure 120). A transmissive approach was adopted, but midway through the explanation the teacher asked if students had any questions. This resulted in a number of questions, for instance about whether variable names needed to correspond between programs, whether constructors could be different for the two classes implementing the interface, and general questioning relating to strategic interfaces. This repeated trial of the same task, interface and activity design as Iteration 2 provides confirmatory evidence that it was the teacher prompting for student involvement that engaged collaboration rather than the interface or the task type.

The screenshot displays the ITEC100 Virtual Classroom 2006 S2 Week 7 interface. The main window shows a whiteboard with four Java code snippets illustrating polymorphism. The code snippets are:

- Exclamation.java**: A class that implements the `Dimensions` interface. It has a `main` method that creates a `Circle` and a `Triangle` object, and prints their heights and widths.
- Dimensions.java**: An interface with two methods: `getHeight()` and `getWidth()`.
- Triangle.java**: A class that implements the `Dimensions` interface. It has a constructor that takes height and width, and methods to return these values.
- Circle.java**: A class that implements the `Dimensions` interface. It has a constructor that takes a radius, and methods to return height (calculated as $2 \times \text{radius}$) and width (calculated as $2 \times \text{radius}$).

The interface also includes a chat window at the bottom left with the following messages:

- Matt Bower**: Hey guys! g'day!
- Matt Bower**: This is a first being first!
- Matt Bower**: Yep. Good to see. How's it all going?
- Matt Bower**: It was nice to have a bit of a break!
- Matt Bower**: Hi guys!
- Matt Bower**: Lost your audio!

On the right side, there is a file share window titled "File Share - Pracwork" showing a list of files:

Name	Size
49543_prac12soln.swf	78 KB
exclamation.zip	537 KB

The interface also features a "Camera and Voice" section with a video feed showing two participants, and a "Attendee List (3)" section showing the names of the participants. At the bottom, there is a "Save To My Computer" button and a "PLAYING..." status indicator.

Figure 120 – Iteration 3 Topic 7 Retest of whiteboard to interrelate source code files

OB2: The tutorial questions were discussed with the teacher broadcasting the solution document using the standard interface and holding a discussion with students about the answers. During the sequence the teacher discovered that if students were granted

“presenter” status (which they normally were) then selecting the “synch” feature did not necessarily synchronize scrolling of the document. This meant that in the past when the teacher had assumed he controlled the point to which the solution document had been scrolled, students had independent control.

OB3: After the break the teacher presented the same whiteboard diagram that had been used in Iteration 2 (illustrating the Triangle class implementing the Dimensions and Colour interface) and discussed this with students. The diagram once again provided students with a visual model upon which they could base their mental model. It allowed the teacher to utilize the multimodal principle (Fletcher & Tobias, 2005) rather than using purely auditory explanations. The diagram was discussed before attempting any of the in-class activities, which was a deviation from Iteration 2 where the diagram was presented after the first question relating to casting had been completed. This allowed the effect of sequencing of instruction to be observed. Students were then asked to complete same two exercises from Iteration 2:

Question 1:

Suppose C is a class that realizes the interfaces I and J. Which of the assignments (1, 2, or 3) require a cast?

C c = . . .;

I i = . . .;

J j = . . .;

c = I; // 1

j = c; // 2

I = j; // 3

Question 2:

Suppose C is a class that realizes the interfaces I and J. Which of the following assignments (1, 2 or 3) will throw an exception?

C c = new C();

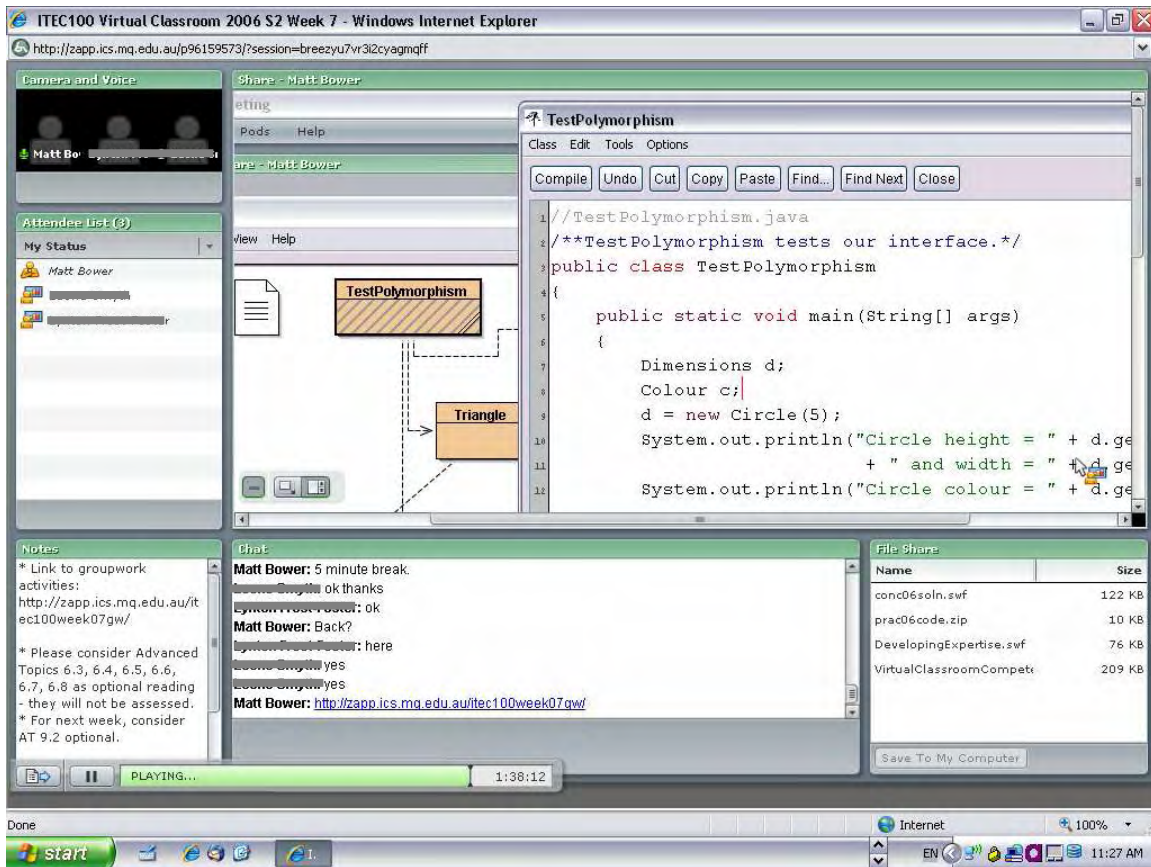
I i = c; // 1

J j = (J)i; // 2

C d = (C)I; // 3

There were only two students in the lesson to attempt this task, providing a very small sample upon which to base conclusions. However, it was observed that providing the diagram before attempting the tasks appeared to develop the students’ understanding and ability to answer the question.

OB4: The task requiring students to add a Colour interface to the original polymorphism example was conducted as a teacher-led programming activity, which differed from Iteration 1 and Iteration 2 where it was an independent activity and group-work activity respectively. A standard sharing interface was used with the teacher broadcasting the IDE (see Figure 121). The teacher-led approach elicited a considerable amount of student questioning and contribution while managing to complete the task in only 19 minutes, again validating the effectiveness of this approach.



OB5: The final part of the lesson involved the teacher presenting and explaining the RandomShape polymorphism applet using a standard screen-sharing interface. Once again the more transmissive approach adopted by the teacher coincided with lower levels of student contribution.

Key Incidents:

- KI1: Asking for questions from students during the transmissive “introduction to polymorphism” sequence resulted in several questions being asked – this is an important way in which collaborative approaches can be integrated into instructive sequences.
- KI2: The teacher realizes how “presenter” privileges effects scrolling of documents. Appreciating this is of critical importance to conducting lessons (to prevent students scrolling through the solution document if the teacher is asking them questions as a means of stimulating collaborations or enquiry). Yet it is in the fourth semester of teaching using the web-conferencing system that this has been realized. This shows how tiny web-conferencing competencies and understandings can have a critical impact on learning.
- KI3: One student’s audio was dropping in and out throughout the lesson which had a significant impact on the quality of collaborations in the class. Technology becomes another potential obstacle to collaborations, not only through the representational forms it enables but also through the reliability of the media itself.

Reflective Notes:

- KI1: Note that the teacher has taken quite a dominant role in this lesson (partially because with such a small class size student contributions were often not so forthcoming).
- KI2: The teacher-led programming approach appears to be a continually effective approach to engaging students while progressing through the material efficiently. The teacher can implicitly (or explicitly) delegate or relinquish control to determine the pace with which the episode progresses, and adjust scaffolding to the ability levels of the students.
- KI3: Independent factors influencing education appear to be either barriers or stimulants. A barrier will block education irrespective of the number or strength of stimulants. Stimulants can only be effective to the extent that there is an absence of barriers. To this extent collaborative technology is actually more of a barrier than a stimulant.

Iteration 3 Topic 8

Topic: Introduction to events and control handling

Attendees: 3

Summary

OB1: To begin this lesson students took turns to present the training module on event handling that they had created. One student's audio was dropping in and out (most likely due to problems with their internet connection) which at times made their presentation unintelligible. Following this the teacher presented the model solution in a highly transmissive mode in order to save time. The entire episode resulted in little interaction.

OB2: As in Iteration 2, the first practical activity for this topic required students to:

*Download a zip file containing two programs,
adjust the RecentreCircle applet so that the mouse click determined the centre of the circle (and not the top left corner)
adjust the ResizeCircle applet so that no matter what the size of the circle the centre remained the same.*

The starting programs were once again shown in note-pods. The task prescription was clearly specified in the top left corner of the interface supporting rapid task commencement. Students were slow to make progress on this task, partially because they found the mathematical concepts relating to centring the circle difficult to understand. The initial teacher prompts and hints did not appear to support students in forming an understanding the concept. In order to provide a clearer explanation of the coordinate geometry underlying the task, the teacher chose to spontaneously use a whiteboard to represent the situation (see Figure 122). The whiteboard allowed audio explanations to be supported by visual representation, leveraging the cognitive gains afforded by the multimedia principle (Fletcher & Tobias, 2005). Students were then able to complete the RecentreCircle task. They were also able to engage with the ResizeCircle task more confidently after the teacher's whiteboard explanation, although still required some prompting and scaffolding from the teacher. The audio modality was easier to use in conjunction with the visual solution space and diagram than the text-chat modality (modality principle, Low & Sweller, 2005) and as such the text-chat pod was not used. The space occupied by the text-chat pod could have been used to house the whiteboard. In total the task took 33 minutes to complete. This was much longer than the previous Iteration, which appeared to be due to the students' ability with spatial concepts more than any aspect of the learning design.

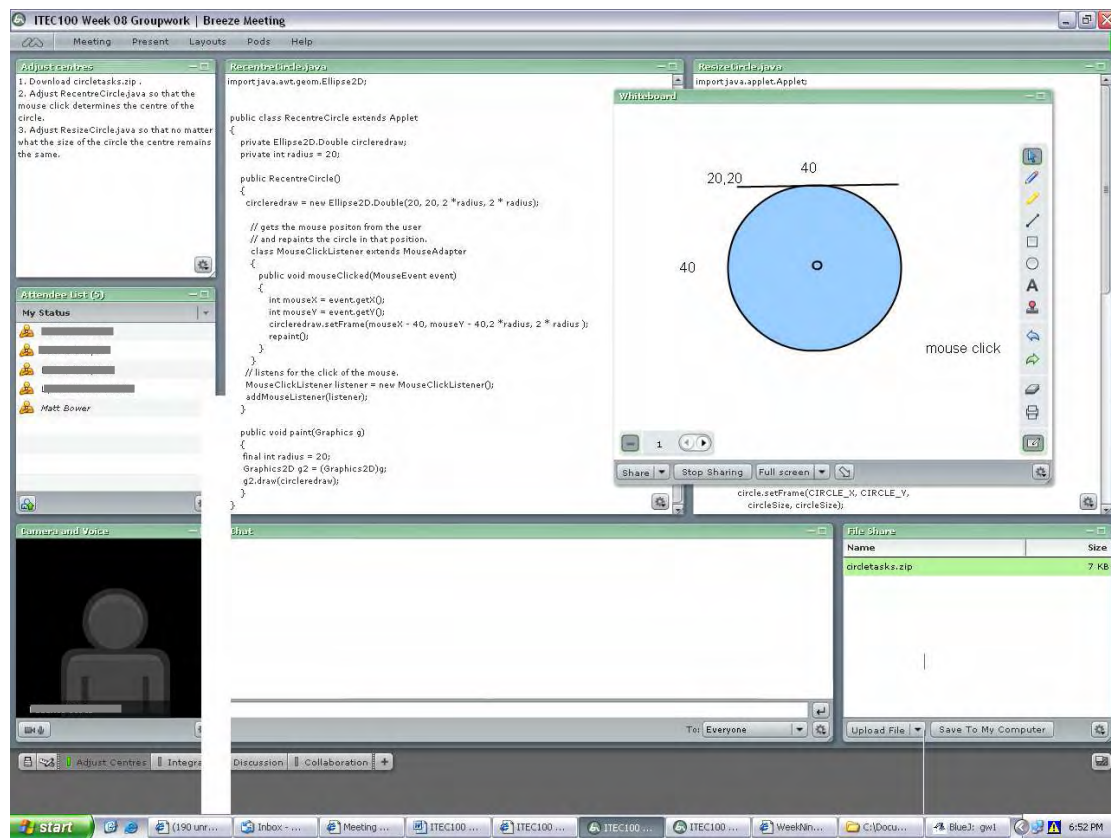


Figure 122 – Iteration 3 Topic 8 Spontaneous inclusion of a whiteboard to support discussion of visual concepts

OB3: After the break the teacher introduced the Integrate task, requiring students to combine the re-centre and re-size functionality of the previous two programs into one applet. The same group-work room interface as the previous semester was adopted, except in this iteration students use audio. In order to address waning levels student contribution the teacher explained that he was going to withdraw from the activity meaning that they would have to interact with one another. The teacher went on to state that if one person made a comment then the other members of the group then needed to provide a response. Establishing the expectation that students take central responsibility for the task improved participation levels. The teacher also set the expectation that students rearrange the room in any way that suited their needs, a responsibility that students readily assumed. One student chose to drag the solution note-pods down so they now covered the text-chat pod. This allowed the pods to be flexibly adjusted to represent their cognitive importance in the learning episode, in accordance with Hollan et al (2000). This change allowed students to see all (or at least the majority) of the program code in the one pod. The effect of this was to reduce both the need to scroll as well as the potential of code being changed by another person without that section of code being in focus (and hence having changes missed by some participants). Covering the text-chat pod utilized the space it occupied that was now obsolete because students in Iteration 3 were using audio rather than text-chat to collaborate. As a further interface amendment, once the relevant code had been extracted from the recentre applet the teacher proposed widening the note-pod containing the Integrate program so that lines of code were not forced to wrap (see Figure 123).

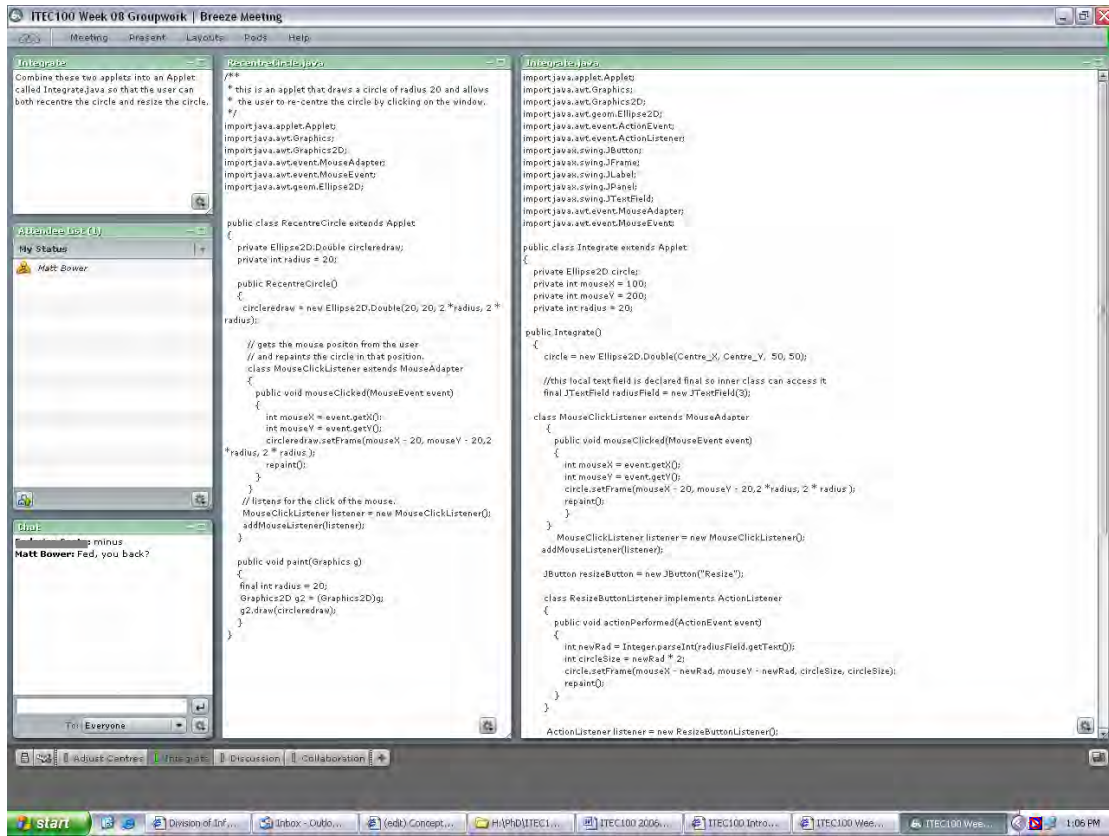


Figure 123 – Iteration 3 Topic 8 Redesign of interface to meet changing collaborative needs

Students agreed to make this amendment to the interface. The use of audio allowed more in-depth conversations to occur than would have been possible using the text-chat because of the speed of contribution it afforded. For instance a discussion was held regarding whether one event listener implementing two different interfaces should be used as opposed to two separate listeners.

OB4: Due to the limited time remaining in the lesson time constraints the teacher chose to broadcast the code for the integrate task on the IDE using a screen-share layout back in the main room. This approach allowed students to choose between watching the teacher debug the program in the main room or make adjustments to the program in the group-work room. The presentation demonstrated how to correctly use the mouseX, mouseY and radius instance fields, which was followed by program testing to demonstrate that the task specification being successfully met. This group programming activity (and eventual completion by the teacher) took approximately 34 minutes to complete.

OB5: The final activity attempted was a reiteration of the changeButton activity from Topic 8 of the previous semester. This was conducted using a teacher-led programming approach (broadcasting the screen and asking for student contributions about appropriate steps to take). The task, interface, and activity design was the same as the previous semester. This resulted in similar levels of student participation and the activity being completed in commensurate time (15 minutes). The use of audio did not noticeably impact on the level of student contribution in this teacher-led approach. Note that before the lesson was closed the

teacher explained how to use the wiki to complete the preliminary conceptual contributions before next week's lesson.

Key Incidents:

- KI1: Technological (connection) problems render the web-conferencing approach to student presentation of preliminary tutorial question solutions useless. Practitioners need to be aware that web-conferencing imposes another layer of potential interference that can inhibit student learning.
- KI2: The use of the whiteboard to represent visual information allowed a clearer explanation of the Circle applet to be provided. Spontaneous use of tools to provide appropriate modalities for the message coincides with greater teacher experience and confidence with the web-conferencing tools.
- KI3: Explaining the collaborative expectations for the Integrate practical task appeared to improve the level of student participation.
- KI4: When students were asked to adjust the web-conferencing interface being used to conduct the Integrate group programming activity they made significant improvements. Offering students the chance to suggest changes to and spontaneously adjust the interface may result in improvements that may not have been considered by the teacher. This draws upon students' experience of collaborating through the interface that the teacher does not possess.
- KI5: The teacher was able to intervene at a critical point in the Integrate task when students were considering implementing two interfaces in the one listener class, and steer them in a more fruitful direction. This was made possible because there was only one group using audio. But with more groups this would have been difficult to intercept, especially since students were not using text-chat and as such no transcript of the discourse could be reviewed by the teacher upon joining the group.

Reflective Notes:

- RN1: Students in the previous semester found the first practical tasks requiring them to augment the recentre and resize circle programs quite simple, whereas this semester students struggled with the task. The teacher observed no major differences in student ability level between the two cohorts, except perhaps mathematical skills. This was a valuable reminder that the pitch of tasks can have a significant impact on collaborations, and providing accurately pitched tasks is entirely dependent on the specific skills of individuals.

Iteration 3 Topic 9

Topic: Abstract classes

Attendees: 2

Summary

OB1: As was the case in Topic 9 Iteration 2, the lesson begins by reviewing the preliminary tutorial question solutions that students have posted on the wiki. Although the teacher screen-shared the wiki and provided a general commentary on the group's answers, the teacher had not prepared specific questions designed at promoting critical thinking and the commentary he provided tended to merely summarize their answers. There was limited student discourse during this review.

OB2: Students were also asked to provide feedback regarding their impressions of completing the conceptual exercises using the wiki. The students indicated that they appreciated being able to build on each other's work in an asynchronous manner. However they pointed out that the first person to contribute may not end up performing any collaboration. While discussing the wiki approach it became apparent that while the "what you see is what you get" input interface was appreciated, one student had not deduced how to use the interface to perform editing tasks (a problem that was immediately rectified via teacher demonstration).

OB3: The model solutions to the tutorial questions were then briefly presented to clarify concepts which students had not described or understood comprehensively. For instance, the parallels between the two ways each of the "super" and "this" keywords can be used within a java file were presented.

OB4: Based on students' weak understanding of the difference between shallow copies and deep copies from previous semesters, the teacher prepared an activity in advance. A diagram of a shallow copy was shown on one half of the whiteboard, and on the other half students were asked to draw a representation of a deep copy (see Figure 124). This activity only took students approximately two minutes to complete, which was much faster than the time taken for the teacher to provide explanations and spontaneous diagrams in previous Iterations. The approach allowed students to be active and productive, and reveal their mental models. From the diagram they provided the teacher could immediately see whether they understood the concept. The fact that a representational form had already been provided for a shallow copy meant that students did not need to spend time discussing how to depict objects and references (as conjectured by Or-Bach & Lavy, 2004). Students agreed that the approach clarified their understanding of the difference between shallow and deep copying.

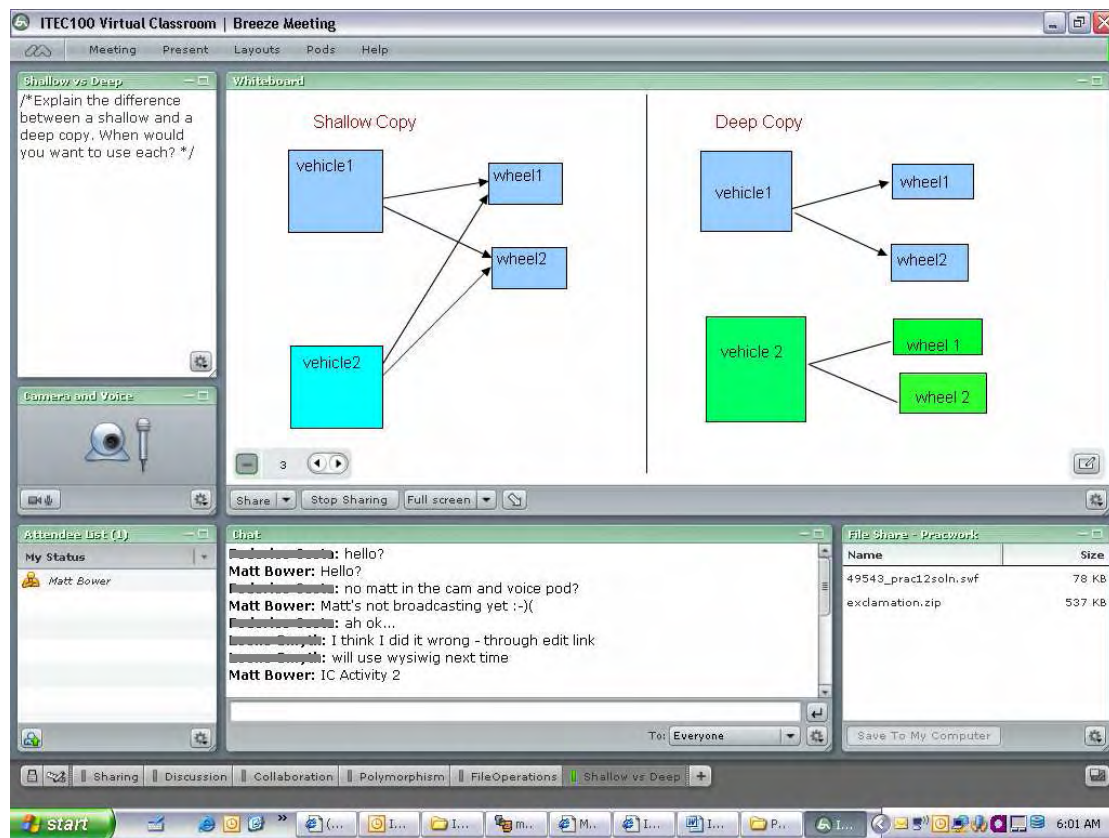


Figure 124 – Iteration 3 Topic 9 Student-centred use of whiteboard to share conceptual information

OB5: Students were then directed to in-class activity 2 requiring them to identify illegal assignments between different superclass/subclass variables.

Question:

Suppose the class Sub extends the class Sandwich. Which of the following assignments are legal?

Sandwich x = new Sandwich();

Sub y = new Sub();

x = y; // 1

y = x; // 2

y = new Sandwich(); // 3

x = new Sub(); // 4

Students were not confident in identifying when assignments were permitted nor the reason why assignments were only permitted in some circumstances. As such the teacher commenced an extended explanation about why subclass references could be assigned to superclass variables but not the other way around. In order to clarify the explanation the teacher decided to draw upon a whiteboard in order to provide a visual representation of how the sub-concepts related (see Figure 125). This once again leveraged the multimedia principle (Fletcher & Tobias, 2005).

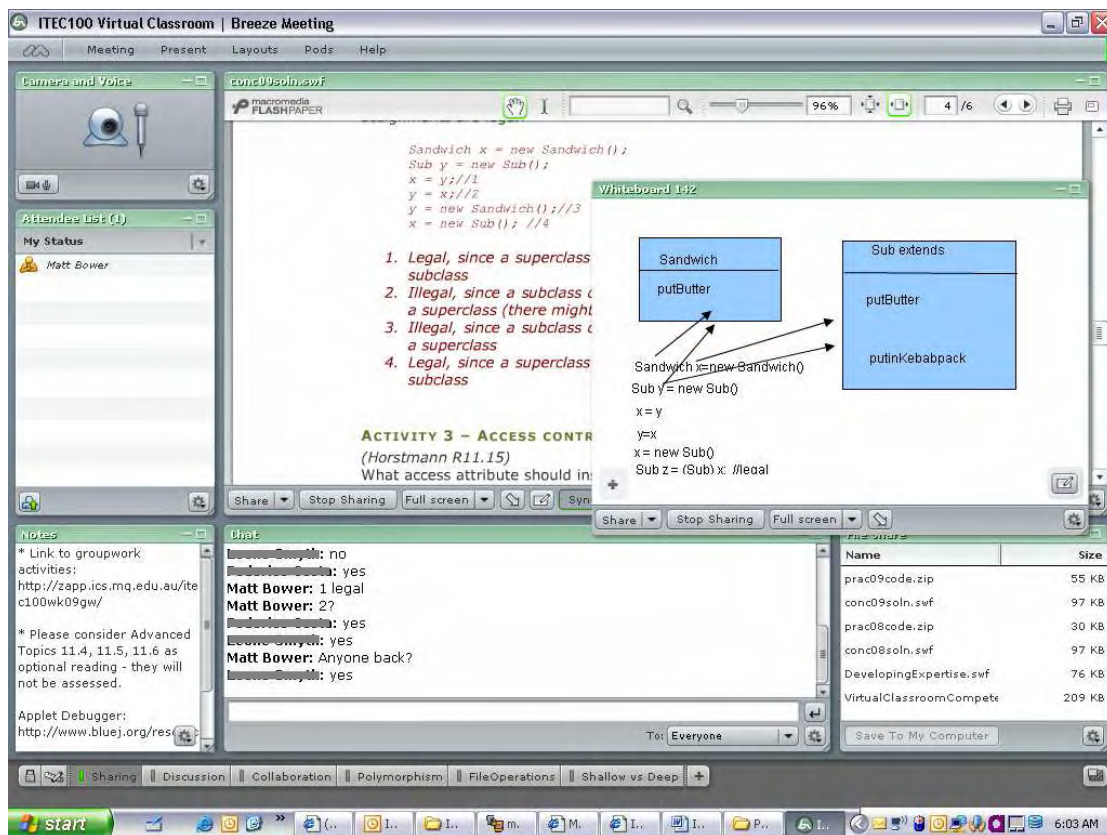


Figure 125 – Iteration 3 Topic 9 Spontaneous use of whiteboard to support discussion of conceptual information

The teacher used the diagram to discuss how the Sandwich (superclass) variable has a putButter method that also becomes inherited by the Sub (subclass) variable. However because the Sub also has a putInKebabpack method a Sub variable cannot refer to a Sandwich reference otherwise it may inadvertently be used to call the putInKebabpack method which won't exist. Explaining this with a diagram allowed the relationships between the components of the concept to be organized. It also reduced the likelihood of student misunderstanding due to cognitive overload caused by attempting to organize too many components of information in their mind (van Merriënboer & Ayres, 2005). Students indicated that this approach clarified the concept for them.

OB6: After the mid-lesson break the students' preliminary practical exercise relating to the extension of the Rectangle class to make a Square class were compared and contrasted. The teacher used screen-sharing of the IDE to navigate through student code and provide a commentary. The quality of the different designs that students had used were evaluated, and improvements suggested. The greater teacher presence in discussions coincided with lower levels of student contributions.

OB7: As in the previous two iterations, the review of the Macquarie University System was the last learning task attempted for this topic. As in previous semesters the teacher dominated the task. However this semester a specific example was chosen for detailed review, and the review had a purpose – to correct the approach that had been adopted by the student. As opposed to the more general review in previous iterations, this provided an authentic task (Herrington, Oliver, & Reeves, 2002). The teacher broadcast his screen

showing the program loaded in the IDE. As the teacher navigated the program the students and teacher discussed problems with the example (duplication of instance fields between superclasses and subclasses, attempting to extend abstract classes from concrete classes, not setting attribute values in constructors). The teacher then rectified these problems in the IDE, at times taking suggestions from the students. Students commented how simple it seemed when the teacher demonstrated how to correct the program. Based on student feedback this implementation was more successful in developing student conceptions than previous iterations. Based on teacher self-perception the approach was more effective in creating a conversational environment (Laurillard, 2002; Waite, Jackson, & Diwan, 2003). The only other activity completed before ending the lesson was an introduction to the major assignment.

Key Incidents:

- KI1: Even though screen-sharing was used to broadcast the preliminary tutorial questions that the students had completed on the wiki, the teacher discourse did not stimulate a great deal of input from students. This may be related to the small number of students in the class, or the broad descriptive commentary (lacking any specific questions for students) that the teacher delivered.
- KI2: The shallow versus deep copy whiteboard exercise provided students with a clear task specification and a representational model that allowed them to complete the activity without needing to hold elaborate preliminary discussion. Using a diagram to dynamically illustrate the difference between a shallow copy and a deep copy provided a clearer representation of students' mental models than textual description.
- KI3: Once again the spontaneous use of a whiteboard to clarify a concept (referencing between superclasses and subclasses) acted as a shared cognitive space, allowing the cognitive load of individuals to be "offloaded to the environment" (Hollan, Hutchins, & Kirsh, 2000). This frees up working memory which can then be used by students to develop their mental models.
- KI4: For the review of the MU system detailed analysis of one concrete example rather than skimming several examples improved the clarity of the description. Setting a specific task objective (as opposed to "conduct a review") promoted student engagement with the material.

Reflective Notes:

- RN1: Using the wiki as the medium for preliminary tutorial question solutions meant that the teacher did not need to organize group-work rooms with material in them – their work is already and always available online.
- RN2: Attempts to draw diagrams on the whiteboard during class are often frustrating because of the poor interface design of the tool (difficult to select and move objects, no copy-paste function, etc.).
- RN3: In many cases the level of student collaboration is mainly influenced by the expectations set and the approach being adopted by the teacher. If the teacher is presenting students appear less likely to contribute.

Iteration 3 Topic 10

Topic: Graphical User Interfaces

Attendees: 3

Summary

OB1: As in the previous week, the teacher adopted a presentational style to cover students' wiki solutions to the preliminary tutorial activities. The teacher broadcast the student solutions (which were brief but well integrated) using a standard screen-share layout and commented upon them. In order to provide an elaborate and well planned answer to question seven (asking them to explain how a miles to kilometres conversion program worked, line by line) the model solution document was presented using a transmissive instructional approach. There were few questions from students resulting with the review being completed within 21 minutes.

OB2: Students agreed to share their preliminary practical exercises and the file-share pod was used to distribute these. The students were designated five minutes to download the files and review them on their own machine. This approach significantly lowered the rate of collaboration, as was observed in Iteration 1 Topic 4 and Iteration 2, Topic 4. After this time the teacher broadcasts his screen with the IDE, describing the background colour changer programs one by one. The interface is entirely teacher controlled. During the teachers' 16 minute monologue there are no student contributions (either statements or questions) relating to the content. The use of a teacher-centred approach more often associated with Iteration 1 also led to a reduction in student discourse in this episode. As well, the review described what students had done but did not focus on supporting students' conceptual development or stimulating discussion. That is to say the way in which the teacher conducted this episode compromised collaboration and learning.

OB3: As in Iteration 2, the group programming activity required students to create a program that uses a JComboBox to change the colour of a panel (based on their previous work that had used radio-buttons). This time the teacher deliberately and explicitly delegated students control over the activity. This resulted in them taking responsibility for discussing concepts and engaging in the problem solving process. The same note-pod based interface as Iteration 1 was used, with students using audio to collaborate. Audio appeared to increase the rate of per-person contribution for this group-work activity, due to the ease and pace with which comments could be made. Once again, the audio mode complimented the textual nature of the code, allowing students to use their dual processing capabilities (Low & Sweller, 2005). Not only did the approach allow students to construct their understanding collaboratively, but the teacher could observe this process to gauge their level of understanding. The "authentic" (Herrington, Oliver, & Reeves, 2002) nature of the task meant that the group required a relational understanding of the content matter in order to achieve the solution.

OB4: After approximately 27 minutes the teacher intervened and suggested that he copy and paste the program they had created thus far from the note-pod to his IDE so that they could debug any remaining errors as a class using screen-sharing. The group agreed to this, and the remaining errors (syntactic) were quickly detected as a team. The whole activity took longer than expected (36 minutes), as the orientation phase (where students were reviewing the

code and devising strategies) took considerable time. However having a task where students are responsible for solving a specific problem and providing them with a collaborative space to do so coincided with a high level of student contribution. This was the last learning episode for the lesson (apart from the final section discussing aspects of the major assignment).

Key Incidents:

- KI1: Once again the teacher covers the students' preliminary tutorial questions by providing a monologue review. The low levels of student input coincided with the absence of specific questions for them to answer or tasks for them to perform. This confirmed similar effects observed in Iteration 1 and Iteration 2.
- KI2: Both the review of the wiki work and the preliminary practical work was full of rich content but lacking in task specification or purpose. This resulted in low levels of student collaboration as well as apparently sub-optimal learning.
- KI3: Some difficulties using the interface using a note-pod for group programming with the long JComboBox were encountered. The program did not fit on the visible portion of one note-pod so the scrollbar was required to navigate to different sections of the code. Therefore it was often not possible to see where amendments were being made by some students. Students thus often needed to describe where they were making contributions.
- KI4: On several occasions the teacher's audio cut out and the students needed to ask the teacher to repeat what he had said. The teacher changed the bandwidth of the room from "LAN" to "broadband" in an attempt to resolve this issue. Students indicated that this improved the audio quality.

Reflective Notes:

- RN1: Sometimes a presentational approach may be adopted by the teacher because it is easier (requires less forethought) and fulfills nominal expectations for the role of 'teacher'. In this case adopting presentational approaches served to confirm that teacher-centred learning designs result in lower rates of student contribution.

Iteration 3 Topic 11

Topic: Arrays and Streams

Attendees: 2

Summary

OB1: This week a more focused approach to covering the preliminary tutorial questions was adopted (as compared to the previous two weeks, for instance). While all questions were addressed by broadcasting student wiki solutions, the teacher chose to cover in detail those questions on which students indicated weakly formed understanding. For instance, for question 3 the teacher copied the solution that a student suggested into his IDE and debugged the program with input from the class (while screen-sharing was on).

Question 3:

Given an array list of Rectangle objects, ArrayList recList and Rectangle rec, write code to count the number of times recList contains rec.

Student solution:

```
public class testRec
{ public boolean check(Rectangle aRec) {
  for (int i=0; i<recList.size(); i++) {
    recList r = (Rectangle)recList.get(i); if (r.equals(aRec))
    matches++
  } return matches; }
} //fc
```

Adopting this approach resulted in more student discussion than the transmissive approaches applied in the previous week. Students made substantial contributions to the debugging process, which drew out several important aspects relating to the use of ArrayLists.

OB2: The more detailed approach to discussing the preliminary tutorial questions was also exemplified in the coverage of Question 6 regarding nested loops.

Question 6:

Draw a diagram that illustrates the arrays that are created by the following program.

```
public class TwoDtester{
  public static void main(String[] args){
    int[][] steps = new int [4][];
    for (int i = 0; i<steps.length;i++){
      steps[i] = new int[i+1];
      for (int j = 0; j < steps[i].length; j++)
        steps[i][j] = i+j;
    }
  }
}
```

Students expressed uncertainty about how to complete this exercise and a lack of understanding about how the code operated. The teacher responded by first discussing with them how they could use println statements to output the contents of the arrays. This was

demonstrated in the IDE using screen-sharing. Students indicated they still did not understand how the program functioned to create that output. This prompted the teacher to lead a machine emulation task on the whiteboard, whereby students had to dynamically represent the state of variables and arrays while they stepped through the program (i.e., their notional machine, du Boulay, O'Shea, & Monk, 1989). This is shown in Figure 126.

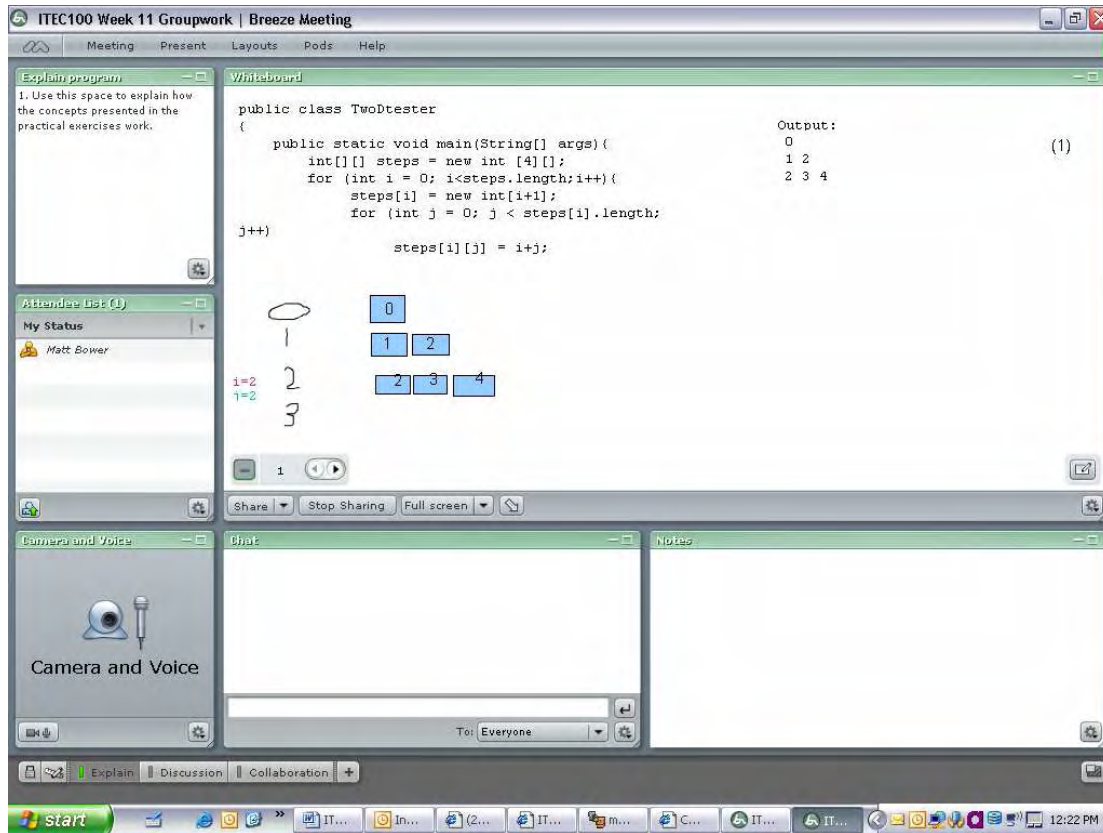


Figure 126 – Iteration 3 Topic 11 using a whiteboard to support guided representation of a dynamic concept

The teacher explicitly set the expectation that students would complete this task. The teacher provided responses to specific questions, ideas for representing programming constructs (array elements, variables), and also provided guiding statements for student activity, all of which supported more efficient completion of the task. At times the teacher also provided extended explanations to support students understand the syntactic and semantic aspects of the code, however students were responsible for determining and representing the state of the notional machine. The level of comprehension that students possessed was immediately evident based on the collective mental model representation they made on the whiteboard. In accordance with Laurillard's (2002) Conversational Framework, based on students' actions on a description of the world the teacher was able to provide feedback, which in turn allowed students to modify their actions (remedy their mental models). The collaborative solution space had allowed the process of student mental model development to be shared, which facilitated greater student insight into the thinking of others and greater teacher insight into the students' learning process. Students indicated the exercise improved their understanding.

OB3: The permutation generator practical task was reviewed by starting with a student's erroneous program. The program was run using the screen-share to demonstrate how it was (incorrectly) repeating certain numbers. The teacher then adjusted the code to function correctly, explaining how it operated in this instance. Students still indicated uncertainty about the underlying logic of the approach, so the teacher chose to again represent the method on the whiteboard (see Figure 127).

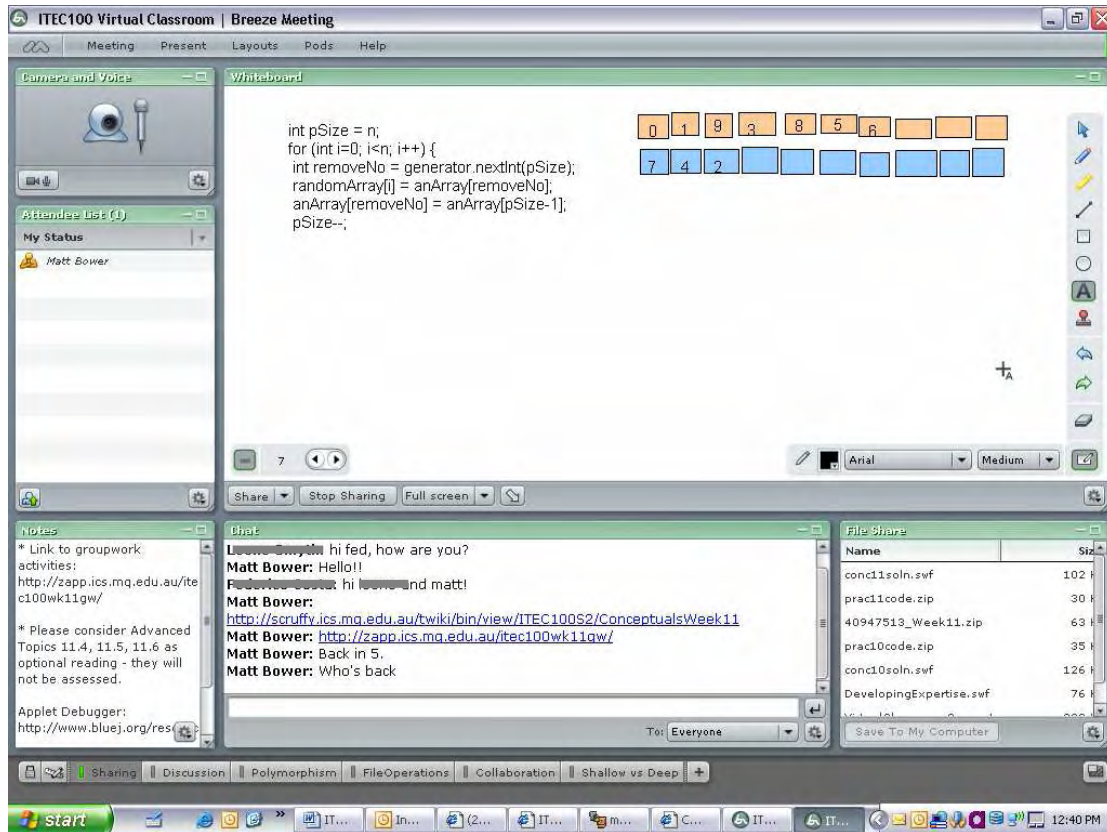


Figure 127 – Iteration 3 Topic 11 Second use of whiteboard to support dynamic representation of conceptual information

The diagram illustrates how an element is extracted from a random position in anArray (originally containing numbers zero to nine in ascending order) and placed in a second array (while the last element in anArray is shifted to the gap created by the extraction). Although using the whiteboard took an extended period of time (almost 28 minutes), the approach allowed the program and general process of using arrays to perform selections to be comprehended by students whereas in previous Iterations explanations had been poorly understood. Students could contribute the next operation in order to ascertain their level of understanding. A dynamic process was able to be modelled. The visual representation leveraged the advantages inherent in the multimedia principle (Fletcher & Tobias, 2005). The numerous pieces of information could be represented and interrelated in a way that would have most likely caused cognitive overload (van Merriënboer & Ayres, 2005) if students were required to follow a purely auditory explanation. The public solution space allowed cognitive load to be offloaded to the environment (Hollan, Hutchins, & Kirsh, 2000).

OB4: Students were asked to download the solution to the final practical task and ask questions if they had any – due to the extended time required to perform a detailed review of

the other activities there was no time to cover all planned activities. Again, the last section of the lesson was dedicated to discussing aspects of the major assignment.

Key Incidents:

- KI1: Debugging the student solution provided for preliminary tutorial question 3 not only allowed that student's partially formed mental model to become completely formed, but provided a concrete example that facilitated formation of correct mental models for other students. Once again making models public (through student's visual representation or discussion of those models) allows the teacher to gauge the level of student understanding.
- KI2: Emulating the program for preliminary tutorial question 6 led to some in-depth questioning from students that resulted in enriching of their notional machine. For instance, one student asked why it was not necessary to specify the number of elements in the inner arrays when the outer array was declared. This led to the teacher explaining that the content of the outer array was merely references to the inner arrays (not the arrays themselves) so the number of elements in the outer array was enough to lay down the data structure in memory.
- KI3: Starting with a student's ill-formed solution and correcting it also allowed the student's mental model for the permutation generator program to be rectified. Correcting student created artifacts provides a relevant task that accurately targets weaknesses in student mental models.
- KI4: The spontaneous use of the whiteboard to represent relationships between programming constructs (in the form of a notional machine) allowed students to understand explanations that in previous Iterations had been misunderstood.

Iteration 3 Topic 12

Topic: Files and System Design

Attendees: 3

Summary

OB1: The teacher chose to briefly cover those preliminary tutorial questions that students seemed to understand, while allocating more time to exercises where student comprehension appeared incomplete or questions were raised. This approach appeared to increase student engagement. Students volunteered ways in which the file-streams material related to their occupational work and asked questions about how the concepts related to commercial practices. Question 8 required students to draw a diagram for the system they had design which is time consuming to upload to the wiki, and so students had not made any diagrammatic contribution to the collaborative solution space. However, one student had submitted a skeleton program for their system design as part of their practical work, so it was possible to look through the program by screen-sharing the IDE.

OB2: Students were asked for feedback regarding the approaches to completing the preliminary tutorial questions that had been adopted throughout semester. The results are summarized in Table 38.

Composition approach	Advantages	Disadvantages
Independent files	Can complete work in own time No timetable to fit into Requires individual to review work thoroughly	No feedback if on wrong track Cannot learn from work of others
Virtual Classroom group-work	Able to use audio to communicate (faster) Some concepts require talking to learn them Learnt more	More time required when first learning to use this approach Difficult to coordinate a common time to meet
Wiki contribution	Could do in own time Most time efficient	Couldn't add comments (as distinct from solution)

Table 38 – Iteration 3 student perceptions of contribution approaches

As in the previous iteration, students indicated strengths and weaknesses for each approach.

OB3: The in-class “grocerybag” activity requiring students to identify inheritance, association and dependency relationships was then attempted using a similar interface that had been applied in Iteration 2 (see Figure 128). The main difference was that students were to discuss the abstractions using audio rather than summarize them in a note-pod.

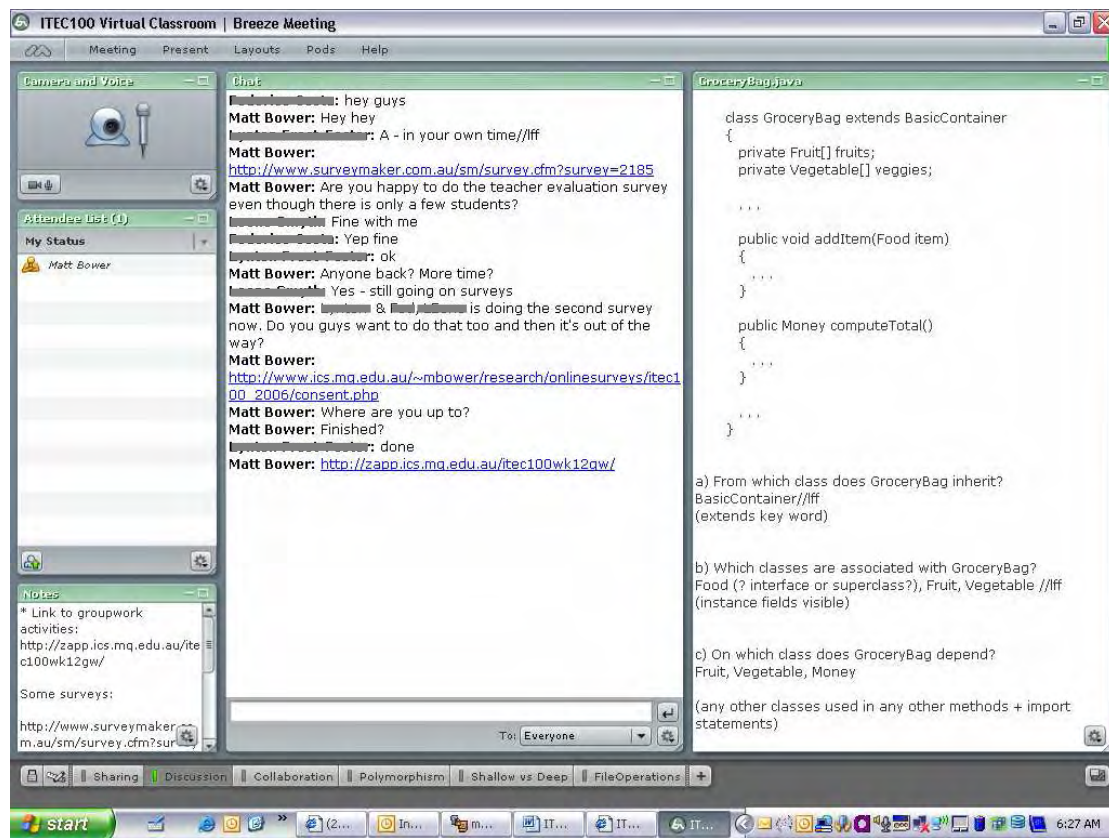


Figure 128 – Iteration 3 Topic 12 Conversational approach to supporting action-process-object cycle

Students were initially confused about association relationships and thus could not identify how they were represented in program code. This resulted in a useful discussion between students and the teacher. Audio once again allowed the group to engage in “tightly coupled” (Neale, Carroll, & Rosson, 2004) collaboration at a pace not possible with text-chat. Following this discussion students contributed answers to the note-pod and were able to summarize (abstract) how relationship types could be identified from program code. The ease and pace of contribution afforded by the audio once again supported more ‘conversational’ (Laurillard, 2002; Waite, Jackson, & Diwan, 2003) approaches.

OB4: Students spent more time than anticipated completing the end of semester evaluation surveys, and so rather than using the whiteboard to compare and contrast student solutions to the exclamation mark practical question (which had been deemed a successful approach in Iteration 2) the teacher discusses students’ approaches to solving the problem with them, and ran a students’ solution in the IDE while broadcasting his screen. This saves time by involving one less transition between layouts.

OB5: The teacher then introduced the next activity by broadcasting the group-work room using the screen-share facility from the main room and explaining the task to students (see Figure 129). To complete the activity students were then asked to edit the program shown in the interface, but students expressed their inability to do so.

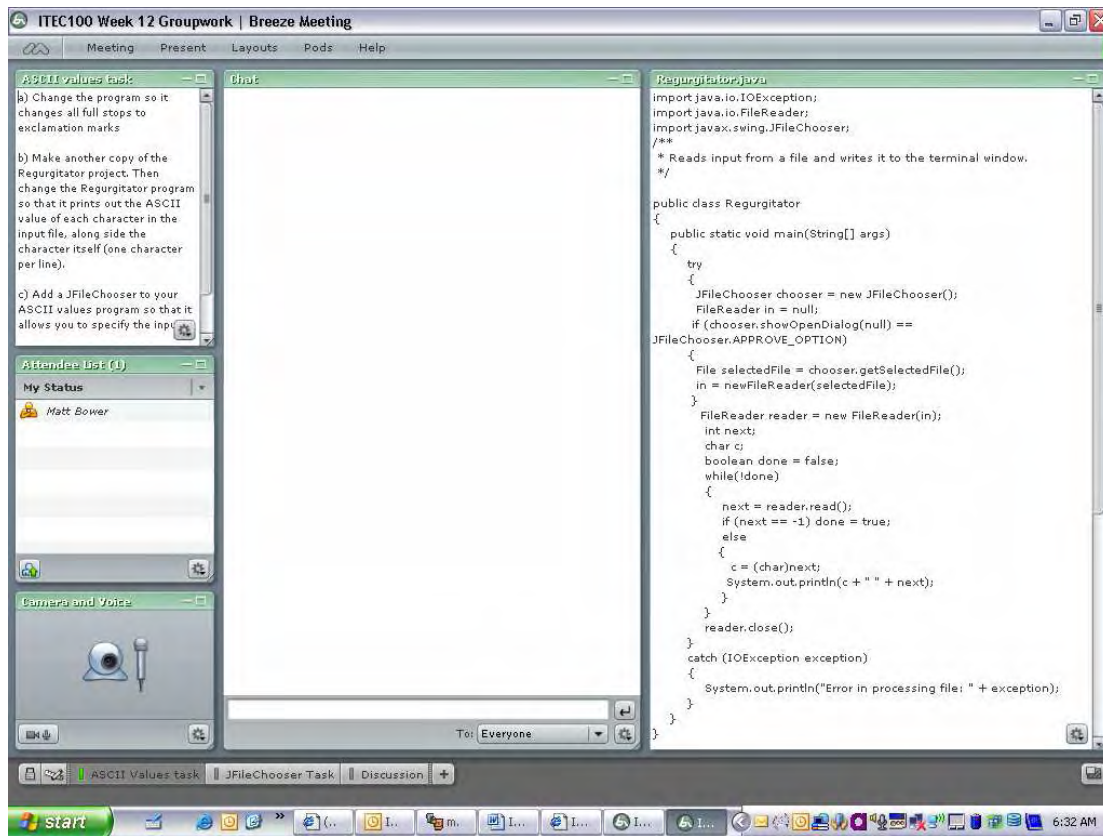


Figure 129 – Iteration 3 Topic 12 Group programming interface

After a minute or so of troubleshooting the class realized that students were trying to edit the teacher's screen-share which was displaying the group-work room (because students were in full-screen mode the screen-share appeared to be the group-work room). The teacher had not directed students back to the group-work room and they had not realized to go there. Once students were in the group-work room they commenced the task requiring them to adjust the previous program solution so that it printed out each character in a file followed by its ASCII number. The teacher left the task for the students to complete, and by discussing the problem amongst themselves students were able to do this within 6 minutes. This task was quite small and required little strategic thinking. The teacher copied their solution into the IDE and tested it for the group while sharing his screen. Note that the chat pod was once again unused because the class collaborated using audio.

OB6: The final group programming activity required students to adjust the file reader program so that it allows users to select the input file at runtime using a JFileChooser. Students were asked whether they would prefer to use a teacher directed approach (with the teacher sharing his screen) or use the note-pod to solve the problem. Students indicated that they preferred the note-pod because they can easily copy and paste into the solution space (be more involved). As well, they indicated that the note-pod provided a more visually clear representation (without movement or orientation issues or screen resolution problems). The activity resulted in some in-depth discussions, with students revealing their partially formed mental models and developing them more fully through their conversation. When close to completing the program, the teacher encouraged one student to broadcast his screen so that they could debug any remaining errors (see Figure 130). The teacher offered guidance on how to accomplish this.

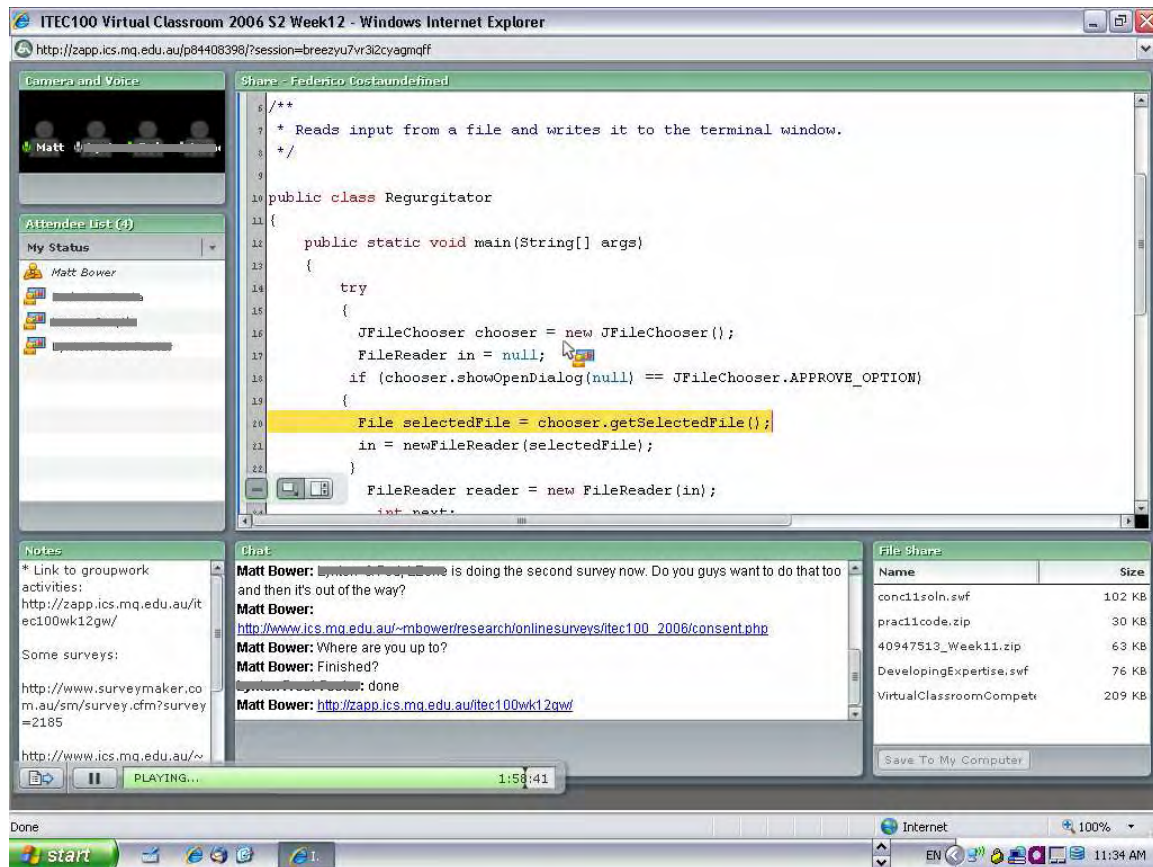


Figure 130 – Iteration 3 Topic 12 Student broadcasting desktop to complete class programming exercise

The task was completed with the student leading the exercise, the group responding, and the teacher contributing occasionally.

Key Incidents:

- KI1: The grocerybag in-class tutorial activity resulted in discussion that supported interrelation of concrete examples and abstract concepts. Note that the interface is clear with no split-attention, students use audio to hold discussions and text to contribute to the solution space, the pitch of the exercise is at a level appropriate for their understanding (they have adequate prerequisite knowledge to attempt the task but learn through the process of completing it). In combination these factors appeared to engage students in conversation.
- KI2: For the exclamation mark practical exercise the teacher considers the fact that transitions between layouts involves some time costs in the lesson due to students moving and orienting themselves to the new space, and so chooses to omit an activity requiring a transition.
- KI3: Confusing a full-screen broadcast of a group-work room for the actual room is an example of a technological competency affecting the quality of the learning episode. This has occurred in the last week of the last iteration of this study, indicating that web-conferencing competencies can be a sophisticated and slow to learn set of skills (for both students and the teacher).

- KI4: The task requiring students to add a JFileChooser to the file-reader program engaged students in more discussion than the task requiring them to change the program to print out the ASCII value of the characters, because it required more elaborate reasoning.

Reflective Notes:

- RN1: The quality of learning discourse appears dependent on the teacher's ability to engage students in a conversation and facilitate the students engaging with one another. A shared solution space appears to support this.