

Tertiary Teaching Excellence Awards 2009

Nomination for

Paul Denny

*Department of Computer Science
The University of Auckland*



Table of Contents

1. Personal Statement	5
1.1. Teaching computer programming	5
1.2. Challenges	6
1.3. Teaching philosophy	6
2. Design for learning	7
2.1. Relevant learning outcomes	7
2.2. Teaching and learning strategies	10
2.3. Student diversity	12
2.4. Encouraging student autonomy	15
3. Facilitating learning	17
3.1. Enthusiasm for subject and learning	17
3.2. Student engagement	17
3.3. Building capability and confidence	18
4. Assessing student learning	25
4.1. Formative assessment strategies	25
4.2. Summative assessment strategies	27
5. Evaluating learning and teaching	28
5.1. Student feedback	28
6. Professional development and leadership	31
6.1. Contributing to effective teaching practice	31
6.2. Commitment to teaching scholarship	32
7. Future directions	33
Appendix: Letters of reference	34



"It is the supreme art of the teacher to awaken joy in creative expression and knowledge"

- Albert Einstein

1. Personal Statement

1.1 Teaching computer programming

My teaching career had a less than ideal beginning. The classroom itself was quite normal – a small 24-seat tutorial room with a whiteboard. By the end of the fourth week of the course more than 40 students were attending. Students sat on the concrete floor, stood around the edges of the room, and spilled out into the hall. As a new teacher I wanted to honour their willingness to learn, but I had no formal teaching training and only enthusiasm and intuition. I felt an obligation to evaluate and develop my teaching ability and effectiveness. As a result, I have pioneered ways to foster communities of students engaged in online collaborative learning, initiated strategies to include and celebrate students with diverse backgrounds, and designed and developed learning software used around the world.

I teach introductory computer science to students in the Faculties of Science and Engineering at the University of Auckland. I have taught in a range of contexts since 1999, including lecturing, computer lab based tutorials, pen and paper based tutorials, and individual tuition. The focus of the courses that I teach is computer programming, which is the process of defining the instructions that enable computers to perform the extraordinary services that are essential to so many aspects of our daily lives. While the science of computing is heavily theory based, the practice is very much an art form, requiring creativity and experience. As a teacher of both the science and art of programming, I support students' along this creative journey, enabling and inspiring them to engage in meaningful ways with content that is highly technical and quite foreign.



1.2 Challenges

Learning to program is difficult. Programming requires incredible precision and patience. Unlike humans, who can make sense of essays containing multiple spelling and grammatical errors, a computer is completely unforgiving – the smallest errors render an entire program unusable. My challenge is to introduce students to a brand new language, one that has its own syntax and rules of grammar that must be adhered to without deviation.

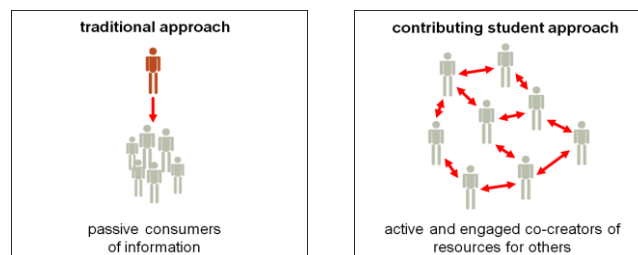
There is no standard high-school curriculum for computer programming and yet many students have vast experience using computers. It is therefore common for students to enter University with preconceptions about the study of computer science. There is often a huge gap between student expectations of what they will be able to do and the dedication that is needed to develop the skills necessary for writing their own software.

I currently teach very large classes, with enrolments usually between 400 and 600 students a semester. A major challenge in teaching any large class, particularly in such a complex subject area, is the enormous range of abilities among the students. Assessments and course activities must be carefully designed to engage novices, but also to challenge and extend the more competent students. Finding appropriate contexts, such as digital image manipulation and interactive game development, that are familiar and authentic to students is critical to engagement.

My teaching is driven by identifying barriers to student learning, and designing approaches and innovations to break these down.

1.3 Teaching philosophy

Involving students in community oriented activities, in which they are responsible for creating and sharing learning resources, is an effective way to promote deep learning. It helps build self-awareness of learning processes and gives students a sense of control over their learning. I integrate activities and assessments into my courses that foster the development of such communities. This kind of engagement emphasizes higher-order cognitive processes such as evaluation, reflection and critical thinking. It helps to transform students from being passive receptors of information, a natural consequence of the traditional lecture environment, to becoming active and critical members of a community engaged in the process of constructing knowledge.



For many students entering university today, using the web to access information is second nature. They are familiar with the value of user-generated content, the lifeblood of Web 2.0. They watch videos on YouTube, and use social networking tools such as Bebo, MySpace and Facebook. As an extension of my philosophy, I have developed an online learning tool, PeerWise, which aims to exploit the familiarity students have with digital technology and social software and engage them in a learning community.

2. Design for learning

I am involved in curriculum oversight at all undergraduate levels and hold several positions relating to curriculum, teaching delivery and the technical support of teaching.

In 2005, as Department Stage 1 coordinator, I led a restructuring of the Stage 1 courses. I began by aligning the learning outcomes with the Association for Computing Machinery (ACM) Body of Knowledge for Computer Science, which is an internationally accepted set of curricular guidelines for undergraduate programs in computing. The lecture content, supporting lab exercises and major course assessments were then developed to support the learning outcomes, using carefully selected and familiar contexts.

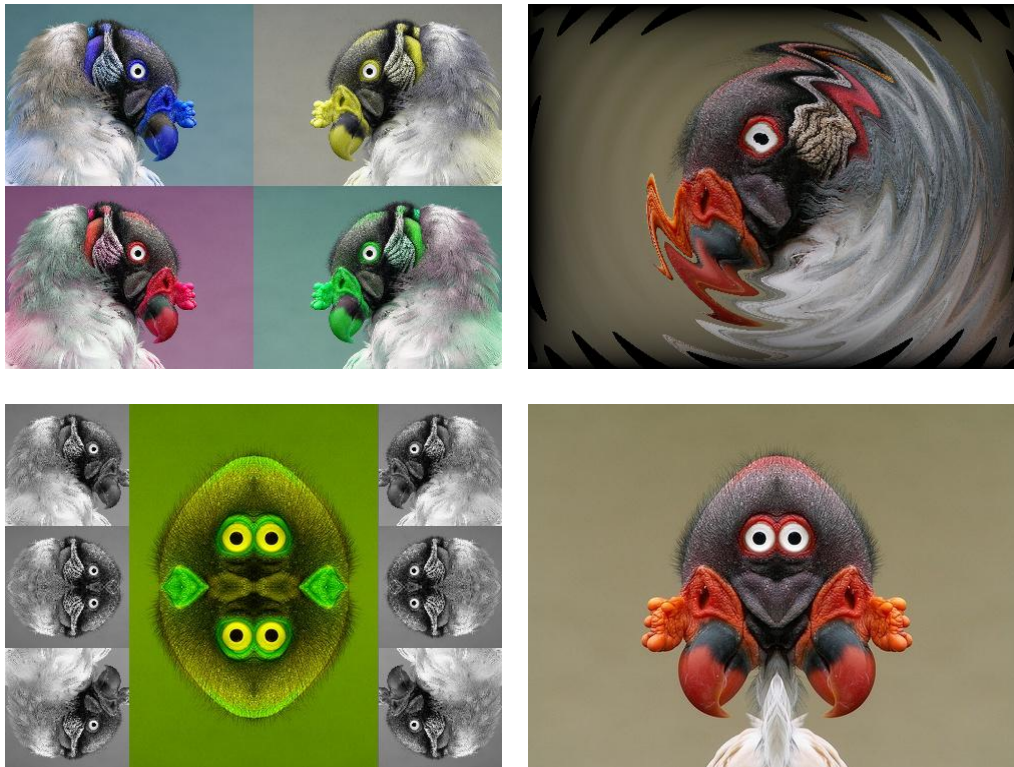
2.1 Relevant learning outcomes

Learning outcomes for the programming courses that I teach are typically technical in nature. An example might be: “be able to write code that uses a nested loop to iterate over the elements of a two-dimensional array”. To introduce such concepts, I find a context that is familiar and of interest to students. In this case, I invite the class to take a digital picture of something they are interested in. We then discuss how images are represented in the computer’s memory, and how they can be manipulated.

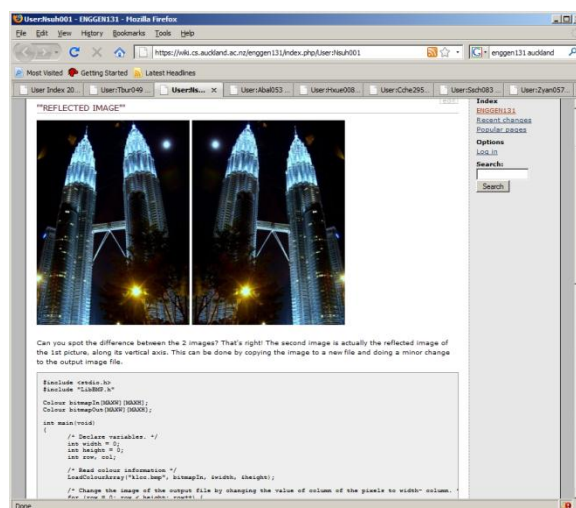


I encourage students to experiment in creative ways to transform the image. In doing so, students actively write code involving nested loops and two dimensional arrays, and can see the immediate results of their work.

Several examples, created by students in 2007, are shown below:



These creative works form the basis for further learning opportunities and community building. Students upload their transformed images to a class wiki for other students to see. They also include a report on their project, reinforcing the concepts they have learnt and serving as exemplars for other students.

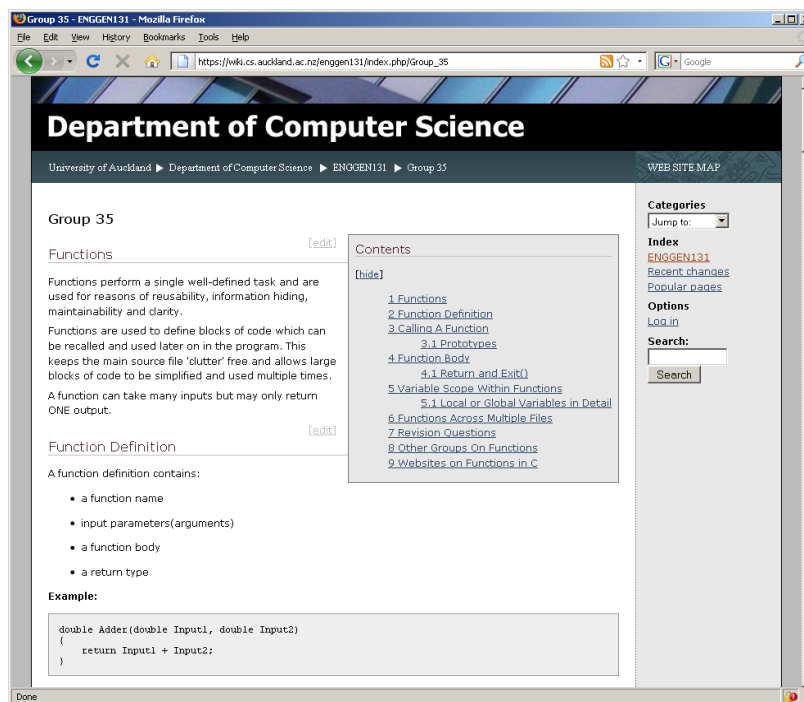


One student's report from the class wiki describing their experience implementing reflection

I showcase many of the creative and interesting transformations in class, and award small tokens of recognition to build student confidence and motivation.

I also assume responsibility for preparing students with skills allowing them to function effectively in today's work environment. The "knowledge economy" demands that students develop skills to work independently, to filter and critically evaluate large amounts of information and to use online tools to communicate effectively. These skills are fostered by the development of a community in which students are responsible for creating and sharing learning resources. Students must reflect carefully on the learning outcomes of a course in order to develop appropriate content. Communication, teamwork and self-assessment skills, all of which are integral to the graduate profile of The University of Auckland, are developed and help build a foundation to support lifelong learning.

One assessment that develops these so-called "soft-skills" is an exam revision wiki in which students work in groups tasked with collaboratively developing revision material on a course topic.



Students learn how to communicate effectively using the "Discussion" page of the wiki to coordinate the efforts of their team. They develop skills in filtering and evaluating the quality of the wiki content, enabling the wiki to serve as a useful revision resource. The community focused aspect of the activity helps to develop a spirit of collaboration and is a central theme of my teaching philosophy.

"Paul has great ideas for making us learn better, eg, the peer review system and the wiki. Thanks."

- Anonymous student feedback, ENGEN 131, 2007

2.2 Teaching and learning strategies

Computer programming is a practical skill. Like riding a bike, no amount of reading or listening to an expert will make up for time spent in front of a computer practising. In 2000, as supervisor for the COMPSCI 101 course, I introduced weekly student laboratory sessions as part of the standard assessment to ensure students practised programming regularly and obtained feedback on their progress. I established the standard curriculum for these labs, matching the content with the learning outcomes for the course, and planning how they would be best organised and run to support lectures. These lab sessions are frequently cited by students in teaching evaluations as being the most useful aid to learning in this course.

The ability to visualise how data are organised and manipulated in the memory of the computer, as a program executes, is critical to successful programming. Students often have difficulty with their program logic, and drawing a conceptual diagram of the computer's memory is an excellent way to tease out their problems. To assist students to develop their own accurate memory models, I developed an interactive online instructional tool, an applet, called CodeStepper.

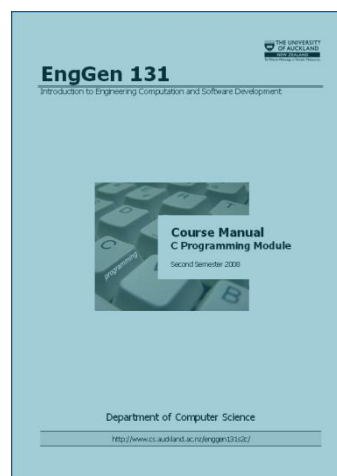
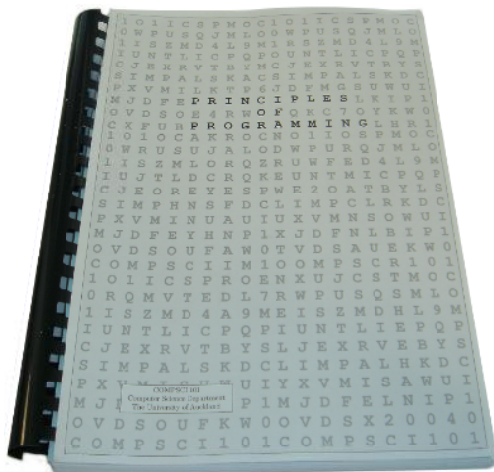
Method calls and parameter passing

The screenshot displays the CodeStepper applet interface. On the left, the 'Code' pane shows Java code with a blue arrow pointing to the line `sum = add(0,0);`. The 'Variables' pane on the right shows two variables: 'sum' and 'difference', both with question marks. Below the code, there is a section for entering initial method parameters for 'add()' and 'subtract()'. A yellow box highlights the 'Enter initial method parameters' section. A text box explains: 'This is an assignment statement with a method call. The value of the first parameter is 0. Similarly, the value of the second parameter is 0. The flow of execution is from the method call to the method definition.' On the right, a 'CodeStepper Question Window' is open, displaying 'Question 3: The max() method of the Math class can be used to return the maximum of two numbers, a and b: Math.max(a,b) How can we use the Math.max() method, in a single statement to return the maximum of three values: a, b and c?' with four multiple-choice options. Option (a) is selected with a red checkmark. Below the question, there is a text input field containing 'Yup - this will work fine! So will (b)'. At the bottom of the question window, there are buttons for 'Answer:', 'a', 'b', 'c', 'd', 'Next Question', and 'Close'. The entire interface is within a 'Java Applet Window'.

Students found this a very useful resource, as evidenced in comments received via an anonymous web feedback form:

"just to really thank you for creating those excellent course applets! as i was panicing for the test tmr, and started viewing them, i realised what a great treasure i've found here~!! yup - u've made my life wayyyyy easier.... thanks!!"

The quality of resource materials has a great impact on student learning. Authoring student coursebooks allows me to include carefully constructed lab exercises that target learning outcomes precisely, and that directly support lecture material and project work. I co-authored the coursebook for COMPSCI 101, and I authored the ENGEN131 coursebook.



These coursebooks are very detailed, collectively over 450 pages, and have been praised by students:

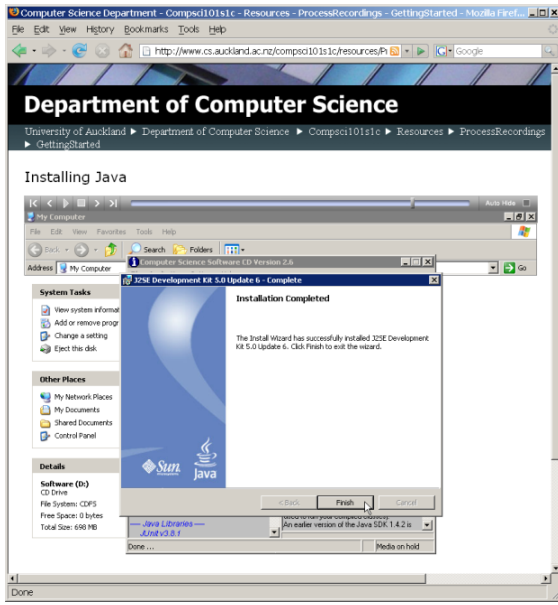
"Incredibly well laid out coursebook, that followed lectures well and gave easy to follow notes during labs and the project."

"The format of Paul Denny's coursebook was fantastic to learn from, easily readable."

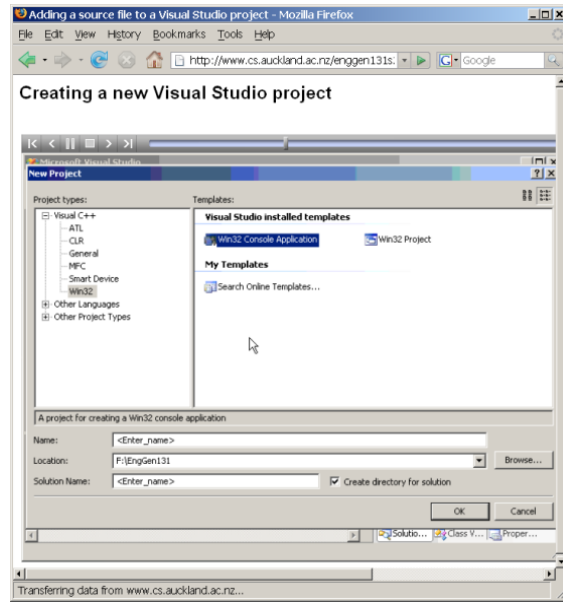
- Anonymous student feedback, ENGEN 131, 2007

To enable flexible learning, I use freely available software in my courses so that students can configure their personal computers without cost and work where it is convenient for them. However, with very large classes it becomes increasingly difficult to provide adequate technical support. Certain technical processes are not well suited to written descriptions, and so I developed a set of screencasts illustrating the process for correct computer configuration.

This set of screencasts was made available online to the COMPSCI 101 course of 2007 and proved extremely useful, being viewed nearly 3000 times that semester (in a class of 538 students) and leading to a dramatic drop in staff time spent helping students with technical setup problems.



Screencast demonstrating how to install and configure the Java development environment



Screencast demonstrating how to configure a Microsoft Visual Studio project

2.3 Student diversity

One challenge that arises when teaching a large, diverse student population is catering appropriately for their range of abilities. I have found that carefully designed open-ended assignments allow flexibility in the solutions, but also remain appealing so that students are interested in what they are doing and encouraged to engage in independent learning.

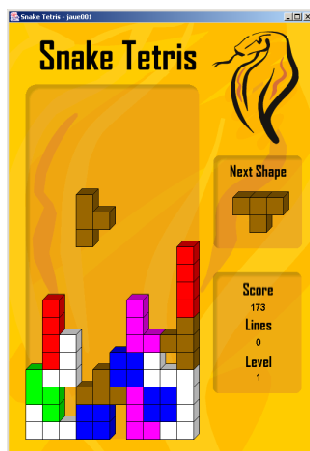
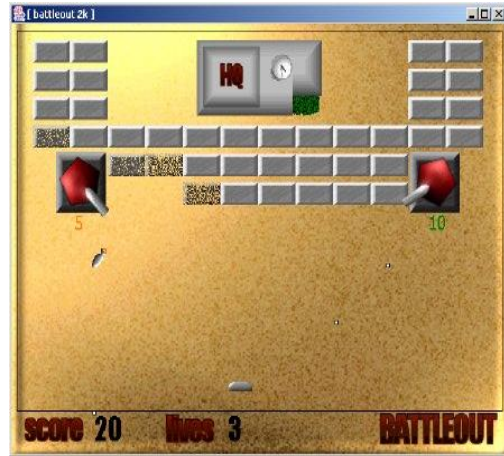
One successful approach that I pioneered in the COMPSCI 101 class is an assignment in which students are given the freedom to develop their own interactive games.

My involvement in the assignment includes providing the software framework, and giving advice to students and helping them develop their ideas, including discussing many advanced topics which are well outside the curriculum.

“His initiatives like the game assignment helped to increase my enthusiasm for the subject. The structure of the assignment gave us the choice to get carried away and generated some really awesome work.”

- Anonymous student feedback on lecturing, COMPSCI 101, 2001

The showcase lecture, in which I demonstrate the submissions to the class, is always very well attended. The quality of the students’ work is often well above the standard expected from first-year students, and is strong evidence of the deep, independent learning which has taken place.



Screenshots of interactive games developed by students in COMPSCI 101

To extend the very top students in class, I organise engaging, voluntary, programming competitions. An example is the Battleships competition for the COMPSCI 105 class of 2004. These students developed the artificial intelligence for a computer player, and their algorithms competed against each other in a round-robin style tournament. To make this possible, I developed the software framework for the main application and the round-robin tournament, and maintained a forum thread for technical support which was viewed more than 5000 times over a 4 four week period.

Not only are these students engaged and learning advanced programming techniques, but they develop a sense of community by sharing a common passion.

“It's these captivating experiences and challenges that started in CompSci101 that have led me to participate, time and time again, in the annual competition that Paul hosts for his students. Though I wasn't always eligible for prizes, the ultimate reward was a series of completed games that I was able to submit as a portfolio to my future employer”

- Former CompSci 101 student, developer at New Zealand's largest gaming company

Another challenge when teaching students with such diverse backgrounds is the necessity for effective pastoral care. I maintain a positive attitude towards all students, and want them to feel comfortable approaching me for help so I publicise my availability in multiple ways. I was a founding member of the Computer Science Department support group, responsible for providing personal support to students in our department.



Support for Computer Science Students

Need to talk to someone?
We are here to listen in confidence and help.

 <p>Paul Denny x87087 Room 467</p>	 <p>Ann Cameron x84947 Room 594</p>	 <p>Santokh Singh x82283 Room 488</p>
 <p>Adriana Ferraro Room C592 x87113</p>	 <p>Linda Thompson x82232 Room 383</p>	 <p>Patricia Riddle x87093 Room 392</p>
 <p>Lei Zhang x88688 Room 395</p>	 <p>Robert Carter x85336 Room 395</p>	 <p>Conny Bluefeld x85720 Room 375</p>
 <p>Andrew Luxton-Reilly x85654 Room 479</p>		

Phone 09 373 7599 followed by the extension number or
visit <http://www.cs.auckland.ac.nz/support-group>

I am chair of the Department Staff/Student Consultative Committee, and oversee regular meetings with class representatives. One of the outcomes of this committee was the establishment of a group workspace for students in a dedicated room with a whiteboard, large desk and several computers.

I have an open door policy, and welcome students at all times.

"I have a young child, so with the time restrictions I have, due to limited child-care, it was always a struggle to find someone available to help. There were many times when I could only seek help in the late afternoon to early evenings. At times, Paul was often finished for the day and ready to go home. Even on these occasions, he still showed a lot of patience, encouragement and a willingness to help out."

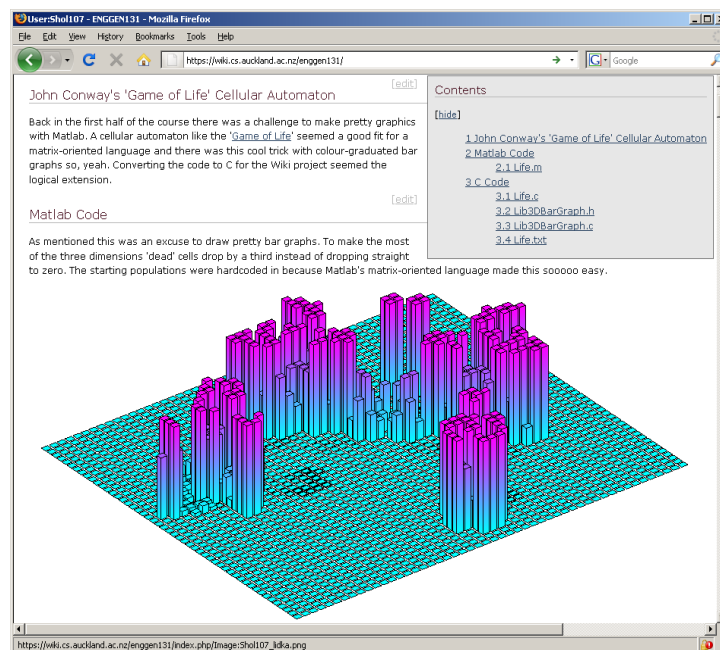
- COMPSCI 101 student, 2001

My care for students was recognised by an award from the Faculty of Science in 2005.

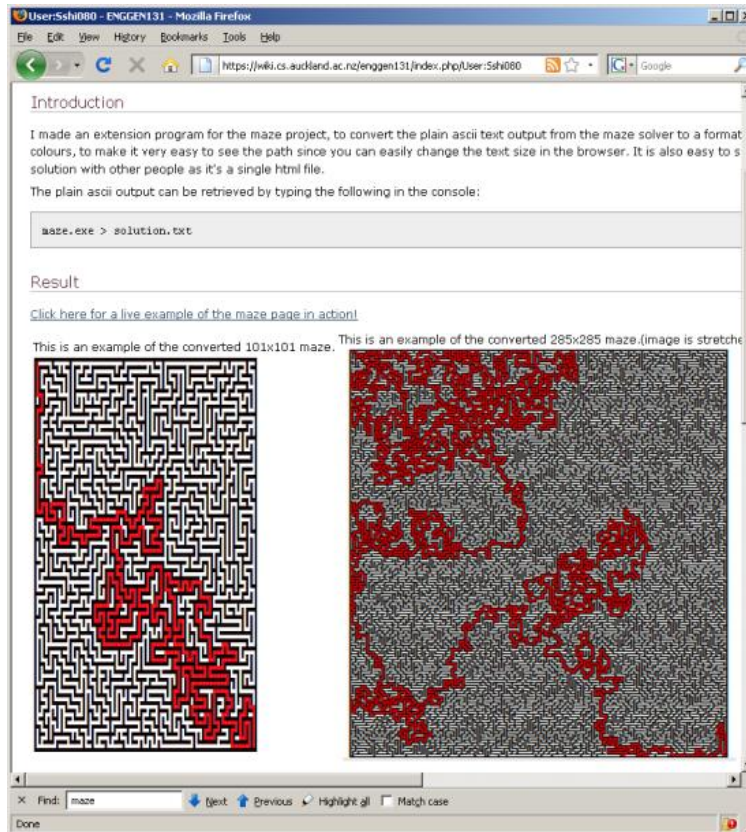
2.4 Encouraging student autonomy

To encourage independence and acknowledge the experience of my students, I design assessments which align with student interests. One example involves them writing, and publishing online, a reflective report about a program they have written. They are free to choose the topic and the scope of their program, in line with their interests and capabilities.

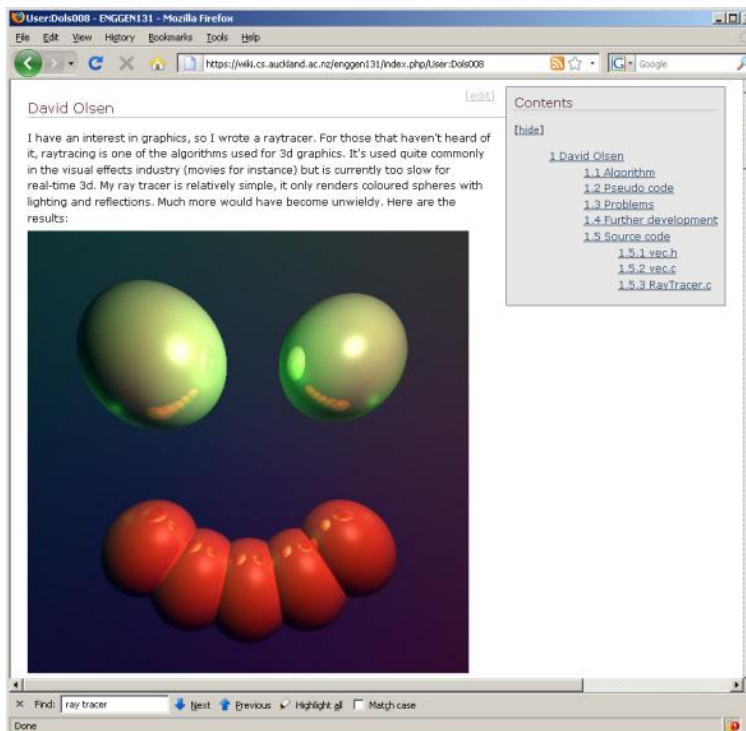
These reports are accessible via a wiki, which exposes students to solutions to a diverse range of problems. The ability to read and comprehend code is an important learning outcome, and this is an engaging and social way of promoting that skill. The creativity and variety produced by students in this exercise is astounding – as shown by the following screenshots of student's wiki reports. Each semester's reports are also archived and available to inspire students in future semesters.



3-dimensional simulation of
Conway's famous Game of Life



Finding a path through a maze



Ray tracer for rendering 3D scenes

3. Facilitating learning

3.1 Enthusiasm for subject and learning

Showing genuine enthusiasm about teaching a subject helps students feel enthusiastic about learning it. Teaching large classes means that often there is more than one stream, and sometimes it is necessary to teach the same class three times in one day. I work hard during every class to infect my students with enthusiasm:

“Paul’s attitude and enthusiasm were the best ever. I never understood so much simply because of somebody else’s enthusiasm for the subject”

- Anonymous student feedback on tutoring, COMPSCI 105, 2000

I am always prepared to interrupt my lectures if I suspect a general misunderstanding:

“He has the ability to continue with the flow of the lecture quickly after questions, interruptions, and the like. Throughout the lecture he requests feedback from the students on your understanding, and will revisit any topic should it be necessary... Paul has without a doubt the most dynamic and modern lecturing style I have attended here at university and other learning institutions”

- Student, COMPSCI 101 and COMPSCI 105, 2002

Learning programming is difficult, and students recognise and appreciate that I understand this:

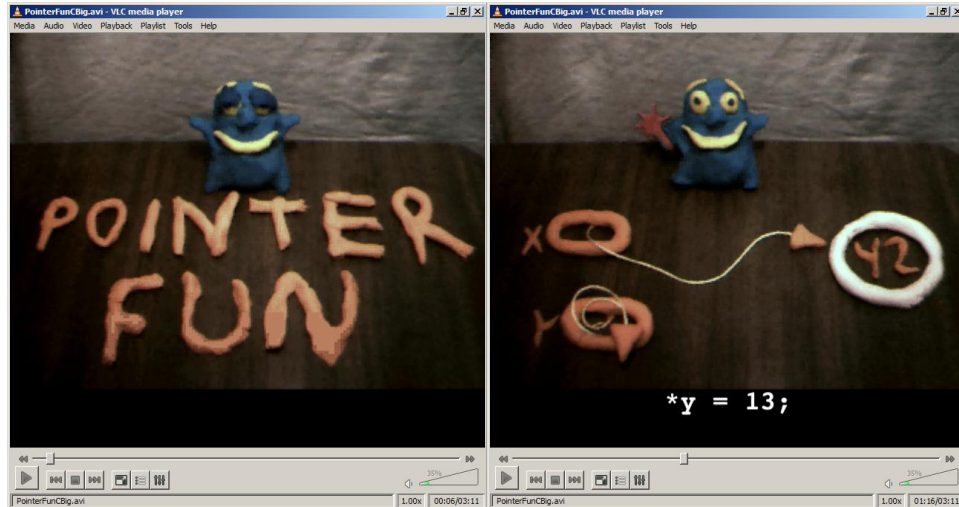
“Having already sat through two degrees of lectures and a teaching diploma (BA, MA, DipTchg)... it is not given lightly when I say that Paul Denny is the best lecturer I have encountered. I’m sure that one day during a lecture we’ll see a sequin-clad assistant and several white rabbits leap from his computer, such is his magic. He conveys an understanding that learning Java is difficult for beginners through his winning combination of humility and humour”

- Student, COMPSCI 101, 2004

3.2 Student engagement

A constant challenge when teaching large classes is to keep students interested and engaged. During lectures, I often take reflective breaks in which I ask concept questions to get students involved and thinking about what we are learning.

I frequently incorporate current, relevant web resources to stimulate interest. An excellent example is the “Pointer fun” video that lets students see the concepts they are learning presented in a fun way. Seeing the material from a different angle reinforces learning.



"Pointer fun" from the Stanford CS Education Library

"Paul has been a lecturer who is unique, distinctive and exceptional... His humour and ability to keep us focused is one of his many attributes that he brings to his daily task of educating us. He challenges students to achieve higher goals by subtly holding competitions or little challenges and games in the lectures"

- two students, COMPSCI 101 and COMPSCI 105, 2002

3.3 Building capability and confidence

Given students' familiarity and passion for using social networking tools, I wanted to provide them with a social and interactive tool for learning, while at the same time developing their communication, evaluation and critical thinking skills.

PeerWise

In November 2006 I began creating a web-based learning tool called PeerWise, which allows students to create, share, critique and answer multiple-choice questions. The ultimate goal of PeerWise was an engaging tool with which students could make individual contributions to a learning resource that, collectively, the entire class would find valuable.

How does it work?

Students are able to create an original multiple-choice question stem, one correct answer with up to four distracters, an associated model answer and a clear written explanation, which can then be answered and critiqued by their peers.

The main menu of PeerWise is divided into three sections: questions contributed by the student; questions created by others that the student has answered; and questions created by others that the student has not yet answered.

PeerWise

EngGen 131 - C (2008 S2)

You are logged in as paul. [Logout](#)

[Home](#) | [Main menu](#)

Your questions

[view](#) You are currently contributing **4** questions
You have deleted **1** of your questions

Answered questions

[view](#) You have answered **6** current questions
1 question you have answered has been deleted by the author

Unanswered questions

All questions [view](#) There are currently **599** unanswered questions you may answer

Followed questions [view](#) There are **0** unanswered questions by authors you are currently following

[View statistics](#) [Provide feedback](#) [Administration](#) [Change password](#)

The main menu for PeerWise

Questions are displayed in tables, with columns listing the date the question was created, how many times it has been answered, and the perceived difficulty and current rating of the question. Students can elect to "follow" authors who have contributed questions that they particularly like. Following an author provides a simple way of discovering good questions, as all of the other questions that author has contributed become easily accessible in a separate section. For authors, attracting followers is an endorsement of the quality of the questions they are contributing.

PeerWise

EngGen 131 - C (2008 S2)

You are logged in as paul. [Logout](#)

[Home](#) | [Main menu](#) > [Your questions](#)

Your questions

You currently have **2** people following your questions
Both of them have answered 1/4 of your questions

You have contributed the following questions :

Questions ordered by date


QUESTION	SUITABILITY	DIFFICULTY	CREATED	RESPONSES	RATING
1 View	OK	easy/medium	4:52pm, 30 Sep	24	3.0000
2 View	OK	medium	1:50pm, 30 Sep	30	3.5455
3 View	!	easy/medium	3:59pm, 23 Sep	18	2.6923
4 View	OK	easy/medium	6:32pm, 19 Sep	96	3.6000

<< Prev | 1-4 | Next >>
(Displaying 1 - 4 of 4)

[New Question](#)

"Your questions" shows the questions written by the student

Every student has access to all of the questions for their course in PeerWise. As the questions are tagged with relevant topics, students using PeerWise for drill-and-practice revision can spend their time answering highly rated questions on topics of interest to them, at a difficulty level they feel comfortable with.



EngGen 131 - C (2008 S2)

You are logged in as paul [Logout](#)

[Home](#) | [Main menu](#) > Unanswered questions

Unanswered questions

You may answer any of the following questions : Questions ordered by rating

QUESTION	SUITABILITY	DIFFICULTY	CREATED	RESPONSES	RATING ↓
1 View	...	medium/hard	1:40pm, 09 Oct	4	4.2500
2 View	!	medium/hard	2:32pm, 03 Oct	27	4.0909
3 View	OK	medium	3:04pm, 03 Oct	47	3.8750
4 View	OK	medium	8:44pm, 04 Oct	23	3.8421
5 View	OK	medium/hard	8:29pm, 04 Oct	8	3.8333
6 View	OK	medium	10:20pm, 29 Sep	46	3.7941
7 View	OK	medium	11:32pm, 29 Sep	48	3.7778
8 View	!	medium	11:55am, 02 Oct	26	3.7727
9 View	!	medium	3:36pm, 29 Sep	22	3.7500
10 View	OK	medium	10:01am, 03 Oct	37	3.7500

<< Prev | 1-10 | **11-20** | 21-30 | 31-40 | 41-50 | 51-60 | 61-70 | 71-80 | 81-90 | 91-100 | 101-110 | 111-120 | 121-130 | 131-140 | 141-150 | 151-160 | 161-170 | 171-180 | 181-190 | 191-200 | 201-210 | 211-220 | 221-230 | 231-240 | 241-250 | 251-260 | 261-270 | 271-280 | 281-290 | 291-300 | 301-310 | 311-320 | 321-330 | 331-340 | 341-350 | 351-360 | 361-370 | 371-380 | 381-390 | 391-400 | 401-410 | 411-420 | 421-430 | 431-440 | 441-450 | 451-460 | 461-470 | 471-480 | 481-490 | 491-500 | 501-510 | 511-520 | 521-530 | 531-540 | 541-550 | 551-560 | 561-570 | 571-580 | 581-590 | 591-594 | Next >>

(Displaying 1 - 10 of 594)






Topics

There are currently questions on the following topics that you may answer (darker topics are more popular):

"1d arrays" "2d arrays" **arithmetic arrays** ascii booleans c casting "command prompt" comment "conditional operator" conditions debugging double errors file **"for loops"** functions **"general theory"** "header files" **"if statements"** input "input output" int **"library functions"** "logical operators" **loops** "nested loop" **output** "peerwise question" pointers preprocessors programmes prototypes purpose "random numbers" "relational operators" security "selection statements" "strand function" steganography strings switch **syntax types** "types of variables" "unary operators" **variables** warnings "while loops"

“Unanswered questions” shows all available questions and topic cloud

Once a student answers a question, they are immediately shown the correct answer suggested by the author, and the number of times each alternative was selected by other students in the course. The explanation provided by the author is also displayed, as are all student comments written about the question.

A	<pre>int i = 0; while (i <= array.length) { i++; // code }</pre>	 5 (4.63%)
B	<pre>for (int i = 0; i <= array.length; i++) { // code }</pre>	 22 (20.37%)
C	<pre>for (int i = array.length - 1; i >= 0; i--) { // code }</pre>	 46 (42.59%)
D	<pre>for (int i = array.length - 1; i > 0; i--) { // code }</pre>	 17 (15.74%)
E	<pre>int i = 1; while (i < array.length) { i++; // code }</pre>	 18 (16.67%)

Once a question is answered, all other student responses are displayed

A range of icons appear, depending on whether the answer selected by the student matches the answer suggested by the question author, and in turn whether this matches the most popular answer selected by other students.



Students are encouraged to use this feedback to make their own judgments about the question content. Rating the quality and difficulty level of each question, as well as providing open-ended feedback, brings students' critical analysis skills into play.

Please rate this question:

DIFFICULTY
Please select what you feel is the most appropriate difficulty level for this question (NOTE: questions of all levels of difficulty are equally valid)

Select difficulty

RATING
Given that you have selected the difficulty level above, please select what you feel is the most appropriate rating for a question of this difficulty to 5 (NOTE: you should take into account the language used in the question, the quality of the alternatives, the quality of the explanation and the overall usefulness of the question)

poor 0 1 2 3 4 5 excellent

Current rating: 2.8364

COMMENT
If you have selected a rating above, then you may also provide additional feedback about this question.

You may choose to write your own comment in the space provided below. This feedback will be visible to the contributor of the question, as well as everyone who answers the question:

Rich text editor toolbar: B I U | List icons | Font family

The rating form for evaluating question quality and difficulty, and for leaving open-ended comments

Students can also choose to agree or disagree with any comments previously written about the question by their peers. An example of a discussion thread is shown below (each small gold star represents one student's agreement with a comment).

WHEN	COMMENT
9:59pm, 08 May	<p>★★★★★★</p> <p>Sneaky. Very good, although it is not how one normally thinks of looping through an array, it is a common pitfall and very well highlighted. Well explained as well.</p> <hr/> <p>Author's reply: Thanks for the feedback! Going through an array backwards might be sensible if the elements are sorted and you want to print them in reverse order.</p>
3:58pm, 09 May	<p>★★★★</p> <p>Good testing of understanding of loops. Awesome.</p>
9:20pm, 08 May	<p>★★</p> <p>while I think the question is quite confusing, this is a great question. (and very great explanation by the way).</p> <hr/> <p>Author's reply: I hope it wasn't too confusing... it was difficult to describe carefully while trying to keep the code to a minimum</p>
12:30pm, 09 May	<p>★★</p> <p>Nice question. A way of looping I hadn't considered until now, but still applicable and within the scope of the course.</p>
9:59pm, 10 May	<p>★★</p> <p>Thinking about the various different increments and conditions which can be used in a loop! I think it's a nice change from the usual loop questions that normally involve an ascending value of i. Brilliant)</p>
5:12pm, 09 May	<p>★</p> <p>Good questions to understand loops and array... Good explanations as well... thank you...</p>

An example of a discussion thread for a popular question

A simple leaderboard is also available, which ranks certain kinds of contributions. This provides each student with an opportunity to compare their performance against that of the most active students.

PeerWise
EngGen 131 - C (2008 S2)

You are logged in as paul. [Logout](#)

[Home](#) | [Main menu](#) > Answered questions

Answered questions

You have answered the following questions :

Questions ordered by rating

QUESTION	RESULT	DIFFICULTY	ANSWERED	RESPONSES	RATING
1 View	✓	medium/hard	10 minutes ago	7	3.7500
2 View	✓	medium	9:01pm, 26 Sep	59	3.7381
3 View	✓	medium/hard	16 minutes ago	11	3.6250
4 View	✓	easy/medium	10:17am, 07 Oct	12	3.5556
5 View	✓	easy/medium	6 minutes ago	9	3.4000
6 View	✓	medium	9 minutes ago	8	3.3333
7 View	✓	easy	10:23am, 04 Oct	20	3.0588
8 View	✓	easy	9:35am, 30 Sep	63	2.7778
9 View	✗	easy	25 seconds ago	96	2.7015
10 View	✓	easy	12 minutes ago	17	2.5714

<< Prev | 1-10 | 11-12 | Next >>
(Displaying 1 - 10 of 12)

Table displaying all of a student's answered questions

Most "answered" question contributor

Total number of answers to all questions contributed by a single user

Total number of answers to all questions you have contributed

RANK	TOTAL NUMBER OF ANSWERS
1	220
2	168
3	137
4	129
5	114

168

TOP FIVE

Most "agreed with" critic

Sum of agreement ratings of all comments written by a single user

Sum of agreement ratings of all comments written by you

RANK	AGREEMENT WITH COMMENTS
1	24
2	24
3	23
4	20
5	18

6

A section of the set of leaderboard tables

Benefits to students

PeerWise offers many benefits to students:

- focusing attention on the learning outcomes of a course
- specific misconceptions must be considered when designing effective distracters for a question
- writing an accompanying explanation for a question requires students to express their understanding of a topic in their own words
- immediate feedback on their answers, and an opportunity to correct any misunderstandings
- many self-assessment and peer comparison opportunities arise – students can see how their answer to a question compares with other students in the class, and can compare the quality of their questions and explanations with their peers
- encouragement to make critical judgments regarding the contributions of their peers, bringing analysis skills into play
- being part of a learning community
- at the end of a course, they have a large repository of questions to use for drill and practice style revision
- a sense of control over their learning

Benefits to staff

PeerWise also offers several significant benefits to staff:

- timely feedback that can shape teaching – common misconceptions can be identified early and addressed in class
- improved student performance – those students who are more actively engaged using PeerWise perform better in final examinations than students of similar ability who are less engaged
- sustainability – PeerWise requires virtually no interaction from staff
- PeerWise is highly transferable, and has crossed both subject and institutional boundaries.

Student led-initiative

The use of PeerWise at McGill University (Canada) was an impressive student endorsement. In 2008 I was contacted by a first year medical student, James Curtis, at McGill. James noticed PeerWise while searching online and felt it could be a useful resource for his fellow students. The Associate Dean of Medicine gave approval for PeerWise to be used and the students have responded by embracing the concept on a completely voluntary basis.

International adoption

PeerWise has proven transferable across disciplinary and institutional boundaries. It has been used in over 10 subject areas in courses ranging from Mechanics to Chemistry in the following institutions:

- The University of Auckland
- University of British Columbia
- McGill University
- University of Melbourne
- University of California San Diego
- Michigan Technological University
- University of California Irvine
- Helsinki University of Technology
- Glasgow Caledonian University
- TAG Magnet High School (Texas)

In the past two years, since the first prototype was developed, PeerWise has been used extensively:

- 6,500 students have participated
- 14,000 questions have been contributed
- 300,000 responses to questions, with evaluation and discussion

Summary

Scalable, technical learning interventions that correlate strongly with enhanced student performance are extremely rare. The fact that PeerWise has been adopted internationally, by a number of institutions, is an exciting result and provides strong supporting evidence of its usefulness.

“Wow, what can I say, PeerWise is one of the best learning tools I have used. The neat thing you have done is encouraged the PeerWise community, in a similar fashion to social networking sites, by allowing feedback to be provided; and utilising a leaderboard. It is the sort of thing that gets people addicted to PeerWise”

- Anonymous feedback from PeerWise survey, ENGGEN 131, 2007

4. Assessing student learning

4.1 Formative assessment strategies

Reflecting on my experiences in teaching large courses allowed me to identify drawbacks in the traditional process for marking projects:

- Feedback is not timely, because of the heavy marking load
- Students do not fully understand the marking process

This reflection prompted me to introduce an additional technique for project assessment. I wanted students to receive more timely feedback, and I wanted to make the summative assessment process more transparent so that all students would be familiar with how their work was graded. I therefore incorporated peer review into this course, facilitated by the Aropä system. When students review a selection of projects from their peers, they gain insight into the marking process and reflect on the way that their own work will be assessed. This has led to a dramatic reduction in the number of re-marking queries that would typically be seen in the courses I teach. Students also receive very timely feedback – within a matter of hours they can read reviews on the project that they submitted.

Aropä
Peer Assessment Server

You are logged in as "A Student".
[Logout](#) [Home](#) [Change Password](#)

Allocations

"Project 2 - C [ENGGEN.131]" (Reviews due by Oct 24, 2008 5pm)

Completed

Allocation 1	View submission	Re-mark	View last mark	View other Reviewer comments
Allocation 2	View submission	Re-mark	View last mark	View other Reviewer comments
Allocation 3	View submission	Re-mark	View last mark	View other Reviewer comments
Allocation 4	View submission	Re-mark	View last mark	View other Reviewer comments

Reviewer feedback

- [Feedback for "Project 2 - C \[ENGGEN.131\]"](#)
- [Feedback for "Project 2 - C - MARKING \[ENGGEN.131\]"](#)

Anonymous surveys and questionnaires

- ["ELECTENG 101 post-activity survey"](#)

The main page for a student using Aropä, displaying the links for reviewing and receiving feedback

Correctness

Compile the source file **from the command line** using the highest warning level (for example "W4" using the Visual C++ Command Prompt):

```
cl /W4 proj2.c
```

Use the data file for marking, called input.txt (available in the submission directory and on the project webpage), and run the program with the **first five** data sets as listed below. Review the following correctness categories:

Compiling

- Code does not compile**
there is at least one syntax error in the code, and therefore the code does not compile
- Warnings**
the code compiles but there are warning messages generated by the compiler
- Code compiles without warnings**
there are no syntax errors, and no warning messages and the code compiles successfully

If the program compiles, then run it and test it with each of the data sets in the test input file.

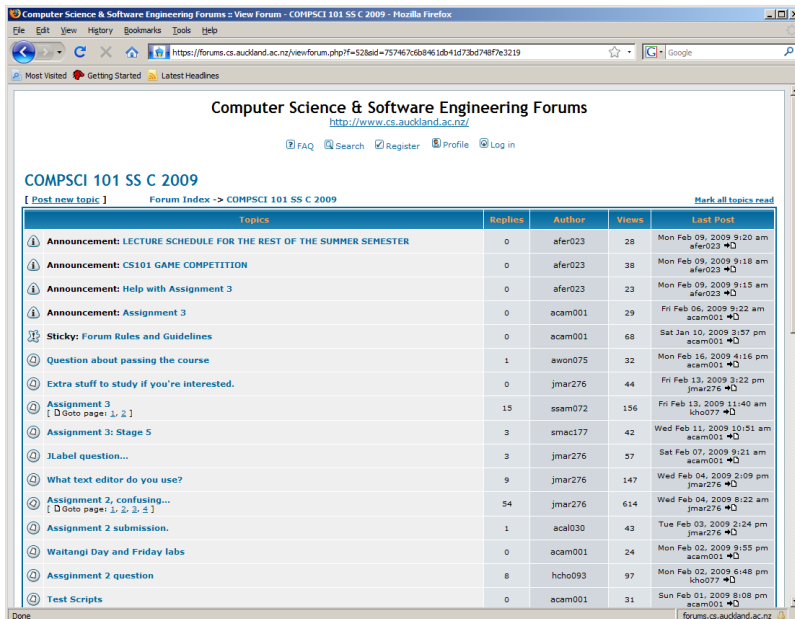
Data set: 1

- Error**
an error occurs when the program runs
- Result incorrect**
the program does not produce the correct output for data set 1
- Result correct**
the program does produce the correct output for data set 1

An excerpt of the rubric I designed for ENGGEN131, that students fill in online

The peer review process is an effective way of exposing students to a variety of programming solutions, and it is contextual as each student has just completed their own solution.

Another tool I incorporate into all my classes is an online forum, enabling students to receive rapid feedback. I moderate the forum, however, mostly students respond to each other's questions, and this act of helping one another builds a sense of community.



Computer Science & Software Engineering Forums
<http://www.cs.auckland.ac.nz>

COMPSCI 101 SS C 2009

Topics	Replies	Author	Views	Last Post
Announcement: LECTURE SCHEDULE FOR THE REST OF THE SUMMER SEMESTER	0	ofer023	28	Mon Feb 09, 2009 9:20 am ofer023
Announcement: CS101 GAME COMPETITION	0	ofer023	38	Mon Feb 09, 2009 9:18 am ofer023
Announcement: Help with Assignment 3	0	ofer023	23	Mon Feb 09, 2009 9:15 am ofer023
Announcement: Assignment 3	0	acam001	29	Fri Feb 06, 2009 9:22 am acam001
Sticky: Forum Rules and Guidelines	0	acam001	68	Sat Jan 10, 2009 3:57 pm acam001
Question about passing the course	1	ewon075	32	Mon Feb 16, 2009 4:16 pm acam001
Extra stuff to study if you're interested.	0	jmar276	44	Fri Feb 13, 2009 3:22 pm jmar276
Assignment 3 [Dosto page: 1, 2]	15	ssam072	156	Fri Feb 13, 2009 11:40 am sho077
Assignment 3: Stage 5	3	smac177	42	Wed Feb 11, 2009 10:51 am acam001
Label question...	3	jmar276	57	Sat Feb 07, 2009 9:21 am acam001
What text editor do you use?	9	jmar276	147	Wed Feb 04, 2009 2:09 pm jmar276
Assignment 2, confusing... [Dosto page: 1, 2, 3, 4]	54	jmar276	614	Wed Feb 04, 2009 8:22 am jmar276
Assignment 2 submission.	1	acal030	43	Tue Feb 03, 2009 2:24 pm jmar276
Waitangi Day and Friday labs	0	acam001	24	Mon Feb 02, 2009 9:55 pm acam001
Assignment 2 question	8	hcho093	97	Mon Feb 02, 2009 6:48 pm lho077
Test Scripts	0	acam001	31	Sun Feb 01, 2009 8:08 pm acam001

4.2 Summative assessment strategies

Computer Science is a practical subject and students work on several projects throughout the semester that measure their proficiency in developing correctly functioning programs.

The correctness of a program can be measured by examining whether it produces the correct outputs for a given set of inputs. Before the project deadline, I provide students with a sample set of test cases, so they can perform some testing on their own. For summative assessment purposes, student programs are graded using a thorough set of test cases, including unusual cases and conditions. Students are given access to these test cases when they perform peer reviews, allowing them to also self-assess their work.

Regular, supervised lab sessions ensure that students keep up to date with course material. To prime students for the exercises in the lab, an assessed “preparation sheet” is distributed which contains a small number of basic exercises. Students work in pairs in the lab on the practical programming tasks, to encourage peer learning and develop communication skills. Ensuring both students have completed the background preparatory work means they will work effectively together.

Students are questioned by a lab tutor when their work is assessed, giving them an opportunity to defend their approach. As I want to encourage mastery of the lab tasks, students are able to correct mistakes identified by the tutor, and have their worked marked again.



5. Evaluating learning and teaching

Reflecting on my teaching is a regular and constant process, in which I consider whether the approaches or innovations I have introduced are effective. Assessing what my students have learnt impacts on my reflection of what I have taught effectively. Changes to my teaching methods are driven by a mixture of this evidence and feedback from students.

5.1 Student feedback

Formal lecturing evaluations include the following questions (with Likert scale responses):

L03: The lecturer was enthusiastic about the subject

L13: The lecturer had a positive attitude towards students

L10: Overall, the lecturer was an effective teacher

The following table shows the average rating I have received (on a 10 point scale) to each of the previous three questions.

Course	Class Size	Semester	L03 Enthusiasm	L13 Attitude	L10 Effectiveness
CS101	356	Sem 1, 2002	9.30	9.33	9.40
CS105	210	Sem 2, 2002	8.83	9.10	8.90
CS101	614	Sem 1, 2003	8.85	9.08	8.72
CS105	323	Sem 2, 2003	9.21	9.18	9.19
CS101	89	Sem 1, 2004	9.51	9.43	9.42
CS101	467	Sem 2, 2004	9.13	9.27	9.26
CS105	266	Sem 2, 2004	9.34	9.43	9.28
CS101	109	Summer, 2005	9.43	9.43	9.54
CS101	448	Sem 1, 2005	9.21	9.17	9.10
CS101	509	Sem 2, 2005	9.18	9.21	9.27
CS105	210	Sem 2, 2005	9.17	9.23	9.00
CS101	472	Sem 1, 2006	9.00	9.05	8.92
EG131	500	Sem2, 2006	9.26	9.34	8.92
CS101	98	Summer, 2007	9.53	9.57	9.65
EG131	575	Sem 2, 2007	9.36	9.43	9.06
CS101	475	Sem 1, 2008	9.29	9.36	9.23
EG131	575	Sem 2, 2008	8.91	8.87	8.84
<i>Total:</i>	6296	<i>Average:</i>	9.21	9.26	9.16

Students have also provided open-ended comments via these evaluation forms:

“WOW. Paul Denny is fantastic. He is extremely knowledgeable and friendly, and has no trouble in getting the class involved in the lecture material. Class interaction is important, and Paul is very good at doing this. I have become very interested in programming as Paul taught the lecture material very well, and at a good pace for the class. Well done Paul :D”

“Paul Denny was truly an excellent lecturer. He made the C Programming section as streamlined and as easy as possible. This is reflected in his outstanding lecturing style, as well as communicating with the students on their level. He made the course enjoyable and full credit to him for doing such a fantastic job”



Response to feedback

Of course, not all feedback is so positive and I comb students' evaluation comments for opportunities to further improve my practice. A frequent request on my evaluation forms was for more worked exercises in class. Programming is a practical subject, done on a computer. Due to the impracticality of holding formal exams in a lab environment, students write exams on paper and wanted to practise more of these kinds of questions in class.

This request prompted me to try a major change in the delivery of my lectures. One semester, I scheduled all Friday lectures as “exercise” lectures, where I would work through example exercises in class. This was not a simple change – the material that would ordinarily have been taught on Friday was incorporated into the earlier lectures in the week – but I felt this structure might give students time to reflect on the material taught that week, and reinforce it during the lecture on Friday.



To evaluate this change, I relied on the feedback students provided at the end of the semester. It was overwhelmingly positive:

“The worksheets we got each Friday were really helpful and it was great that we could go and ask questions”

“Excellent interactive Friday lecture system really helped understanding of the weeks lectures”

These Friday “exercise” lectures are now a regular feature in this course.

6. Professional development and leadership

6.1 Contributing to effective teaching practice

Sharing my teaching methods and ideas with colleagues, both within the University of Auckland and more widely, is one way I can contribute to effective teaching practice.

From March to June this year, I will be visiting the University of British Columbia in Vancouver to share PeerWise with Carl Wieman (Nobel Laureate, Physics, 2001) and his group, the Carl Wieman Science Education Initiative, and evaluating it in an international context.

In 2008 I co-authored a paper describing the Java framework I created for supporting the development of interactive games for motivating students to engage in independent learning. I presented this work in March at SIGCSE 2009 (The ACM Special Interest Group on Computer Science Education) in Tennessee, where it was well received.

"I enjoyed learning more about your project and look forward to adapting it for my courses. Please send the source code and any sample projects you can provide"

- personal email correspondence
- [name removed], School of Computing & Info Systems, Grand Valley State University and session chair, "Graphical Frameworks", SIGCSE 2009

Techniques I have used to support student learning have been adopted by other departments. The ENGGEN 131 course is co-taught with the Department of Engineering Science which is responsible for half of the course. In 2007, staff from Engineering Science adopted the class revision wiki activity after seeing its effectiveness in the 2006 course.

I am actively involved in peer mentoring and helping new tutors and lecturers adjust to the department, from providing moral support to helping in the design of lecture material and assessments. I attend lectures given by new staff, and provide advice and constructive comments on their teaching resources and lecturing styles.

"I have worked in the same teaching team as Paul since I first started as a tutor in 2000. Paul is simply the best colleague and mentor that anyone could hope to have. He has provided constant support, encouragement and guidance to help me develop my teaching skills. Paul is a brilliant lecturer and a very kind, caring person. He is a superb role model for other tutors and lecturers, and gives his time generously to help colleagues to improve their teaching delivery and to develop more effective teaching materials. I feel very proud and honoured to be part of Paul's teaching team as he is so high regarded by both students and staff"

- [name removed]
- Senior Tutor, Computer Science

6.2 Commitment to teaching scholarship

I have an ongoing commitment to personal development and teaching scholarship, and have attended and presented at conferences and workshops in my discipline.

A paper describing the efficacy of PeerWise won the best paper award at ICER 2008, the premiere conference in Computer Science Education research. This award identified the paper likely to have the biggest impact on the discipline. This presentation was pivotal in the wider adoption of PeerWise at institutions overseas.

I am a member of the BRACElet project, which is a longitudinal multi-institutional, multi-national study that aims to provide insight into the domain of reading and writing code as a novice programmer. The project has been underway for four years and consists of over 30 academics from 16 institutions in five countries. As part of this group, I have been involved in published research on a new technique for assessing programming ability in written exams. We investigated a technique, known as a “Parsons problem”, both qualitatively and quantitatively and have shown it offers insights into a student’s understanding that are not offered by more traditional exam questions.

I am an active member of the Computer Science Department’s Education Research group. The group is a venue for sharing ideas on teaching and assessment. Findings from our work are disseminated to colleagues through the regular series of Software Engineering Research Group (SERG) seminars. These seminars allow me to reflect on, and communicate my work to others.

I recently attended a workshop on contextualized approaches to CS education at the Australasian Computing Education conference to learn how to further exploit students’ familiarity with digital media when teaching computing.

I regularly attend the University of Auckland Teaching and Learning showcase. In 2008, I presented on PeerWise and its effect on student learning to a full lecture theatre. This has helped to raise the awareness of PeerWise at the University and led to an article being published in the “*aCADemix*” University newsletter.

I have received teaching awards at both the faculty and university level:

- Faculty of Science Dean’s Award for Distinguished Teaching (2004)
- The University of Auckland Teaching Excellence Award for Innovation in Teaching (2008)

7. Future directions

In the immediate future, I will continue to share my work on PeerWise with colleagues throughout the institution, New Zealand and further afield. I am committed to furthering the impact of this educative tool in diverse contexts.

I am currently in the early stages of developing a new tool which I hope will provide an engaging way for novice programmers to learn and think about programming. It will be a tool that will excite and stimulate even the most hesitant learner to grapple with computer programming in relevant contexts.

Preparing this portfolio has highlighted for me that while not all ideas I explore have been or will be successful, investigating them and evaluating their effectiveness is of paramount importance. It has reinforced for me the value of finding relevant contexts to ground course content and motivate students. I will continue to create opportunities for learners to take active roles in communities collaboratively engaged in meaningful learning. This engagement builds metacognitive skills that promote self-awareness of learning processes, and develops foundational skills to support lifelong learning.

Teaching remains my passion, fuelled by my own love of learning. I know that what I strive to do, in my lectures, through my course materials and through the tools I design to reduce the barriers to learning, enables students to appreciate the art of computer programming.

Computers are tools that we must learn to use creatively for the betterment of society, and students in my classes are awakened to this responsibility and challenge.