

Thinking and Behaving Scientifically in Computer Science: When Failure is an Option!

Anne Venables and Grace Tan
Victoria University, Melbourne City, Australia

Anne.Venables@vu.edu.au Grace.Tan@vu.edu.au

Executive Summary

In a Finnish study of four different academic disciplines, Ylijoki (2000) found that in Computer Science there was a disparity between the conceptions held by undergraduate students and staff about their discipline; students viewed it as being far more pragmatic and results focused than did their instructors. Not surprisingly, here at our Australian university where the undergraduate Computer Science program emphasizes programming and problem solving skills, the authors had noticed a similar inconsistency between staff and student beliefs. This paper reports on an effort to realign these conceptions and broaden student experience using an assessment task. Centered on solutions to the popular 'Sudoku' puzzle (Sudoku, 2005), the task was designed and introduced into an Intelligent Systems course, a final year elective of a Computer Science degree. The goal was to expose students to some of the 'pure' rather than applied aspects of the Computer Science discipline (Becher & Trowler, 2001), by using assessment to encourage experimental learning (Kolb & Fry, 1975).

The assessment specification instructed students to design and conduct several 'in silica' Computer Science experiments to solve and/or create Sudoku puzzles. Importantly, students were asked to keep a Research Diary documenting their thoughts, attempts, backtracking and progresses as they attempted the assignment. Most unique from a student's perspective was that 'failure' to solve the given problem by experimentation was a viable option; their efforts would be rewarded given they conducted themselves 'scientifically' in their attempt.

The inclusion of a Research Diary as part of the assessment proved beneficial to both students and staff, particularly as the Diaries focused students on the doing, rather than on the results achieved. For staff, these journals allowed the privilege of seeing what students were thinking and doing as they engaged in the problem. An early analysis of Diary entries shows evidences of students 'thinking scientifically', that is, understanding and thinking about the problem before formalizing an approach to a solution. In addition, Diary entries also show evidences of students 'behaving scientifically' as they reported conducting experiments and analyzing their results, before concluding about their experiences. For students, regardless of whether they able to complete the task, the Research Diaries were valuable testaments to their efforts and experimentations.

Keywords: Experimental learning, Scientific thinking, Scientific behavior, Research Diary, Sudoku puzzle

Introduction

In their sociological research into academia in the UK and the US, Becher and

Material published as part of this journal, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

Trowler (2001) describe the knowledge landscape as being measured in two different dimensions, one being a continuum between the soft and hard disciplines, and the other graduation being between pure and applied academic pursuits. Within this framework, engineering and computer science occupy academic territories in the quadrant described by the intersection of the hard and applied dimensions, thus they are purposeful, pragmatic disciplines that focus on technique development and production. More specifically, Ylikoki (2000) found in a Finnish study that “the disciplinary culture of the computer science tribe is professionally oriented, emphasizing the virtue of hard expertise wanted by computer firms in business life”. Not surprisingly, these descriptions resonate with the study of Computer Science here at our university in Melbourne, Australia.

In line with the described disciplinary culture, our undergraduate teaching curriculum places a strong emphasis on the acquisition of skills and expertise, such as programming and problem solving. Additionally, like many other institutions (Upchurch & Sims-Knight, 1997), the capstone task in our degree is a software engineering project designed to solve a particular business problem for an industry client. With such an applied and practical focus, it is perhaps not surprising that we have noticed a disparity between staff and student beliefs similar to that reported in the Finnish study mentioned above. There, of the four disciplines examined “*Only in the case of computer science is there a difference between teachers’ and students’ conceptions, since the former try to steer the disciplinary culture in a more academic direction and the latter in a more business life direction*” (Ylikoki 2000, pg 395). This paper reports upon the introduction of an assessment task designed to realign these conceptions by broadening the student experience. By deliberately shifting some of our teaching along the axis between the pure and applied (Becher & Trowler, 2001) towards the pure sciences, we hoped to expose students to some of the ‘pure’ aspects of our discipline and to encourage them to recognize and appreciate the scientific mindset within Computer Science.

The vehicle chosen to do this was the elective study of Intelligent Systems in our undergraduate degree. This course is an introduction to the various problem solving strategies and heuristics of artificial intelligence covering such diverse topics as expert systems, robotics, machine intelligences, genetic algorithms and neural networks. Typically students attempt this elective in their third and final year since some programming and software development experience is required to contend with the broad scope of the subject. By introducing an assignment task where students were asked to explore a problem solution by conducting several (Computer Science) experiments, we hoped to shift student focus more onto the process rather than the solution, i.e., towards the ‘pure’ end of the computer science spectrum and away from the applied. To achieve this, students were asked to maintain a Research Diary during assignment completion where they recorded their thinking and behaviors. Even more unique from a student’s perspective was that ‘failure’ to solve the given problem by experimentation was a viable option; their efforts would be rewarded given they conducted themselves ‘scientifically’ in their attempt.

Rationale for & Design of the Task

Although much academic argument continues about the true nature of science and the scientific method (Chalmers, 2003; Popper, 1972; Stewart, 1995), in their advice for research students Phillips and Pugh (2000) point out there are two quite separate aspects of the scientific approach. The first aspect is that of the classic hypothesis testing and deductive logical approach, recognizable to most readers as the use of experimentation to decide between possible alternate hypotheses or problem solving strategies being under investigation. Ideally, this experimentation is systematic, repeatable (Stewart, 1995) and is usually described in a scientific report; classically divided into sections titled Aim, Method, Results, Discussion and Conclusions. A good scientific report is ordered and logical and reflects upon the prior experimentations of predecessors and places the current experimentation within the known body of related work. The second and often ignored as-

pect of the scientific approach is the psychological behaviour that brings about the report (Phillips & Pugh, 2000 pg 18). Importantly, it is all the guesswork, inspiration, correction, blind alleys, hypothesis testing, dead ends and backtracking that is done when exploring a problem scientifically.

With both aspects of the scientific approach in mind and the knowledge that there is considerable empirical evidence to support that student learning is strongly influenced by assessment (McDowell, 1995; Miller & Parlett, 1974; Synder, 1971) the authors set about designing an assessment task to promote both scientific behaviors. Guided by the epistemology of constructivism and its recognized place in science education (Abraham, 1982; Lorschach & Tobin, 1992), it was important that the task allowed students to experience active science, i.e., conduct experiments, so that they could use their experiences, successful and not, to build their own internal knowledge. Experimental learning as previously described by Kolb & Fry (1975) as a cyclical process passing through the four stages: experiencing, reflecting, concluding and testing also emphasizes the integration of theory with practice (Kolb, 1984). As such, the importance of active learning experiences for studying science is recognized and recommended by the National Science Education Standards of the US (National Academy of Sciences, 1995).

The assessment task specification asked students to design and conduct several “in silica” Computer Science experiments. To commence, students needed to link classroom and textbook theory to practice and apply their programming skills and the theoretical knowledge of problem-solving strategies and heuristics. They were encouraged to research possible solutions from any available source, to discuss their approaches with classmates, to reflect upon the outcomes, to conclude and conduct further testing. Completion of all of these steps would encourage and support students to adopt a deeper learning approach (Biggs, 2003; Biggs & Collis, 1982), one where students compare, relate and synthesize their research together with their experiences. In addition, students were encouraged to work in pairs rather than alone, so that they could discuss their ideas and strategies.

As educators we wished to know if our students would be thinking and behaving scientifically in their assessment attempts and this desire compelled the need for a Research Diary to be completed throughout the duration of the task. The notion for the diary came from the practice in physical science laboratories of keeping a daily record of experiments. In this instance, such a journal was to detail items such as the students’ first thoughts upon receiving the assignment, their progresses both forward and backward, and their reflections upon outcomes of experimentation with different solution strategies. As well, students were encouraged to voice their frustrations with setbacks and their joys at successes. In fact, the Research Diary would in itself be a ‘work in progress’. It was hoped that it would be more than just a data and record of design notebook as used by others (Upchurch & Sims-Knight, 1997) and more akin to reflective journals most often used in the humanities (Borg, 2001).

Implementation

Critical to the assessment task was the selection of suitable problem for students to solve experimentally. The problem needed to have both depth and breadth, in that it could be approached using one of several different strategies. Equally important, to allow ample possibility for experimentation, was that none of these approaches should be obviously correct or incorrect. There has been a long association between the study of game playing and their solution strategies in Artificial Intelligence and Intelligent Systems courses (Jones, 1987), and puzzles and games have been often used to improve teaching in computer science (Feldgen & Clua, 2004; Jenkins, 2001; Ross, 2002) as they are particularly suitable for assigned projects because their exploration is less straightforward. Additionally, as Epstein (1999) points out “*To play the most difficult games well, however, a program must contend with fundamental issues in AI: knowledge representation,*

search, learning, and planning”, so the problem chosen for ‘scientific exploration’ was the increasingly popular Sudoku puzzle (Sudoku, 2005), shown in Figure 1.

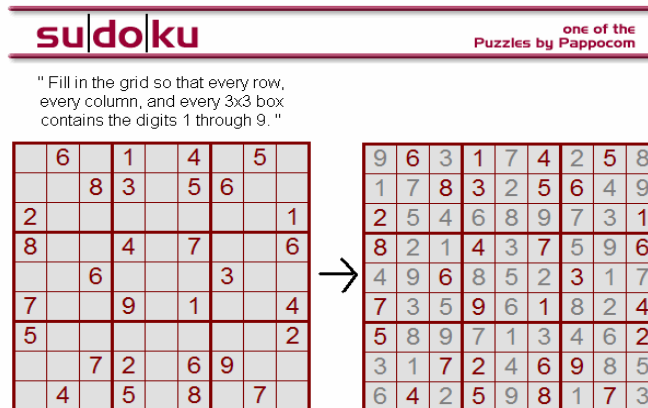


Figure 1: A Sudoku puzzle and its solution.

Sudoku is a puzzle of 81 squares where each row and each column of the puzzle contains the digits 1 to 9, as does every smaller 3 * 3 grid. To solve it, requires no mathematics, other than the ability to count from 1 to 9, but logic, reasoning and intelligence (natural or artificial) is needed to solve a Sudoku puzzle. It is a problem open to exploration by various approaches, some examples being genetic algorithms and rule-based systems. As well, the representation of a Sudoku puzzle within a computer program requires some consideration as the choice of an appropriate data structure impacts on the way manipulations are carried out when solution strategies are being applied. In introducing the assignment, as suggested by Giguette (2003), a small class exercise illustrating the basic steps of Sudoku was run. This exercise proved helpful to students in understanding the puzzle and what they would be attempting to accomplish for their assessments.

Ordinarily in Computer Science, a typical assessment task focuses on the creation of a software solution to the set problem. In this instance, as seen in Table 1, no specific instructions were given on how to solve the problem and the production of software was considered as incidental. Rather students were asked to write a scientific report of their search and coding efforts. They were encouraged to concentrate on the process with the unusual instruction that “failure IS an option AS LONG AS it is well described”.

Table 1: Part of the Sudoku assessment specification

Grading for this assignment will be weighted towards the report of the experiment over the production of an intelligent system. Therefore students who investigate the problem thoroughly will be rewarded regardless of whether a viable solution can be found.	
In this case, failure IS an option AS LONG AS it is well described.	
Your report will be structured using the classical scientific notation detailed below	
Report section	Should cover
Aim	<p>Make a clear statement about what your plan is for your assignment/experiment.</p> <p>Do you plan the strategy to</p> <ul style="list-style-type: none"> • solve a Sudoku puzzle given certain squares • generate Sudoku solutions for 81 empty squares • do both of the above

Method	<p>Make a clear statement about which strategy you choose to use for your assignment. It is suggested that you either use a</p> <ul style="list-style-type: none"> a) rule-based expert system OR b) genetic algorithm OR c) hybrid(combination) approach <p>The method should include</p> <ul style="list-style-type: none"> • A detailed description of the steps taken to try and achieve the stated aim using the chosen strategy. • A list of sub experiments and details of how each was done
Results	<ul style="list-style-type: none"> • Results of each sub experiment (please include failures as well as successes) • Overall result of all the experiments
Discussion	<p>Explain</p> <ul style="list-style-type: none"> • Why each step or sub-experiment was undertaken • What each sub result revealed • What the overall result means • Any plans for future work on this problem
Conclusion	Sum up your experiment(s) and what you found
References	Any references you used to help run these experiments
Appendices	<p>Copy of code created, JESS, Java or both</p> <p>Any results of (successful/failed?) experiments additional to those covered in the results section.</p> <p>Instructions on how to run your intelligent system (if created)</p>

Students were assigned two goals. The primary goal was to conduct a set of Computer Science ‘experiments’ that explore a strategy to either solve or (and) produce Sudoku puzzles with the secondary goal being the creation of an intelligent system based upon the results of these experiments. The grading of the assessment was structured so that production of the scientific report of experiments, with the possible inclusion of a software solution, accounted for the 80% of the marks. To encourage exploratory learning and experimentation, the remaining 20% was assigned to the submission of a Research Diary. So regardless of how students meandered across the solution space, there was academic currency and reward in documenting the process. It was hoped that the assessment of the Research Diary would further underline to students the importance of the process and in its completion they would experience some of the psychological behaviours that bring about such a report (Phillips & Pugh, 2000).

Results & Discussion

Victoria University, Australia is a dual sector university that caters to an ethnically diverse student body through both onshore and offshore programs. At the home campus, located in Melbourne, students undertaking this third year computer science elective of Intelligent Systems form a diverse and heterogeneous group; locals share learning experiences with new migrants, mature aged students study alongside high school graduates and industry experienced students assist their computer novice classmates.

For this assessment, the somewhat counter-intuitive inclusion of a Research Diary had two unexpected benefits. Firstly, it forced students to do some earlier planning and the thinking about their task particularly as the specification was deliberately loose and open-ended. The second side effect was that there was no detected incidence of plagiarism amongst the submissions as students simply acknowledged any assistance given. Upon deadline thirty-nine groups handed in a submission, of which only twenty-two found a working strategy with fewer groups producing a vi-

able software solution. Yet the majority of submissions were assessed as a passing grade when there was evidence in their reports and Research Diaries that they had conducted an adequate amount of scientific investigation. Thirty-six groups submitted a supporting Research Diary where the dated entries allowed us the privilege of seeing what our students were actually thinking and doing at various stages of the assessment completion. We report here upon the first analysis of these diaries and comment on their collective content.

For instance, student first impressions were often recorded, such as

“An assignment on a game! Great!” and

“Sudoku is good mental calisthenics but very addictive.” and

“Started solving puzzles and became addicted!”

Such comments add support to our supposition that the current worldwide popularity of the Sudoku puzzle would be a useful vehicle to motivate and engage students in the assessment task. Although a necessary precursor to student learning (Greening, 2000; Knowles, Holton, & Swanson, 2001), such interest is not evidence for scientific thinking or behaviors that we hoped to encourage. As iterated by Stewart (1995), to conduct science one must be able to think scientifically, that is, state the problem to be solved and to formalize a systematic and rational approach. Later in the Research Diaries, we see evidences of students attempting to formalize an approach to the problem, such as one entry

“Tried to solve Sudoku manually but couldn’t complete. Gave the puzzle to other friends to solve the puzzle manually. Try to find out more from the Internet. Was successful in completing the puzzle manually....”

... .. Discovered that there was a pattern in the puzzle, so attempted to reversed the pattern and tried to solve it again”.

Also, associated with this mindset are scientific behaviors where one will conduct experiments and examine results critically to decide between alternate hypotheses. Another student wrote

At first I tried this way.....(diagram) ... but , as is readily apparent, although this version obeys the sudoku rules for rows and columns, it has duplicate numbers in each and every 3x3 square ... This successfully worked around the problem.”

Many different scientific behaviors were reported, for instance

“Exploring on the net again, we got the idea to simulate the puzzle in MS Excel and thought about trying the same strategies in Java or JESS.”

This group tried and made a Microsoft EXCEL version that added up separate rows and columns. In testing this version students realized that their solution did not check for repeat numbers and was therefore incorrect, necessitating a revision of their approach. Similar experiences of failures and backtracking were reported by other groups. Normally, such ‘failures’ and the lessons learnt are not submitted as part of a standard assessment, yet they are important steps to understanding a problem and drive re-experimentation to find a solution. Without the submission of a Research Diary, we would not have an insight into students’ thinking and behaviors. Most importantly, through the diaries we also witnessed student synthesis of their results.

“Initially, we planned to solve the puzzle with the genetic algorithm approach however, whilst creating and experimenting with code in JAVA we came upon some coding errors, prompted us to change our solving strategies to rule-based and logic algorithms.”

Having discovered some evidence of students’ behaviors of hypothesis testing, backtracking and problem exploration, we decided to take a closer look at the diary entries. Each was searched to

identify evidences in support of the students exhibiting ‘scientific thinking’ and ‘scientific behavior’. Two milestones of ‘scientific thinking’ were identified as *Understanding and thinking about the problem* and *Formalizing an approach to the problem* and a list of evidences for these as reported in the Research Diaries is given in Table 2. Likewise for ‘scientific behavior’, the two milestones *Conducting the Experiment* and *Analyzing the Results* were identified along with the associated evidences as well. Each instance of a milestone was recorded, thus the reported counts of instances may be greater than the total number of Research Diaries analyzed.

Table 2: Evidences for scientific thinking and behavior and their instances as recorded by 36 student groups in their Research Diaries.

		Milestone	Evidence	Instances
S C I E N T I F I C	T H I N K I N G	Understanding and thinking about the problem	Reading the assignment carefully Searching and playing Sudoku puzzle found <ul style="list-style-type: none"> • Online <ul style="list-style-type: none"> ○ Official Sudoku website ○ Other relevant websites and forums • Print materials such as newspapers, magazines, staff newsletter, books Discussing with others with/in <ul style="list-style-type: none"> • The group • The lecturer • Outsiders such as friends and acquaintances • Electronic chat forums 	48
		Formalizing an approach to the problem	Searching for a systematic approach to solve the puzzle Thinking about the various AI paradigms in particular those covered in lectures Thinking about formal programming constructs in coding Developing an algorithm or pseudocode.	40
	B E H A V I O R	Conducting the experiments	Attempting to <ul style="list-style-type: none"> • Code the puzzle in Excel using standard functions • Code a solution algorithm using rules and the rule-based expert system shell, JESS • Use brute force number crunching in JAVA • Code a genetic algorithm solution using various programming languages • Use paper based explanation for their experiments • Mimic researched solutions found online 	55
		Analyzing the results	Reporting of successes and/or failures of attempts Attempting to rationalize the outcomes with the body of knowledge previously gathered Rethinking about the approach Improving upon the approach	54
		Concluding about the experience	Reporting a shift in thinking Reporting enjoyment of the experience	36

Given the specification as shown in Table 1, where students were asked to find solutions experimentally, it is not surprising that both hallmarks of Scientific Behavior, *Conducting the Experi-*

ment and *Analyzing the Results*, were recorded more often in the student diaries. However to arrive at these behaviors, we argue that students needed to be ‘thinking scientifically’, i.e. *Understanding and thinking about the problem* and *Formalizing an approach to the problem*. With only twenty-two groups finding a possible solution strategy before deadline and with fewer groups producing a viable software solution, we are not surprised that *Concluding about the experience* recorded fewer instances. We are particularly encouraged that several groups documented shifts in their thinking, like

“To solve Sudoku puzzle, logical decisions have to be made to change a number in a different spot according to other fields in the grid! That concept works; it’s completely impossible for a GA to solve!?”

And the following from one Research Diary

“Genetic Algorithms (GAs) cannot effectively solve problems in which there is no way to judge the fitness of an answer other than right/wrong, as there is no way to converge on the solution. These problems are often called “needle in a haystack” problems.”

“You don’t need to be a genius to figure out that a GA stinks for solving this puzzle.”

“My challenge is to make an AI that’s better than a brute force algorithm. It’s already taken my algorithm 10 minutes to get alright results and it has only happened in that time period once out of 10 runs. I’m not leaving my computer on for 10 years just to get a solution.”

In fact, with permission to fail in their attempts several groups reported upon the enjoyment of experience

“Although we didn’t create an intelligent system, we are happy with the research and experiments we conducted in order to come up with our rules to create and solve sudoku puzzles during this strenuous assignment.”

“Although we failed to carry out the task we did enjoyed the effort and fun in attempting to solve this mind blowing puzzle!”

Conclusions

On reflection, the Research Diaries were an observable and quantifiable record of student thinking and behaviors documenting their experiences of Computer Science that are closer to the ‘pure’ end of our academic discipline. The diaries contained evidences of backtracking and rethinking about a problem and displays of scientific behaviors such as conducting of experiments to test hypotheses and reflective thinking about the outcomes.

But, although students recorded that the Sudoku problem was motivating and the task was engaging, it is reasonable to ask: “Did students actually learn anything from all this ‘thinking and behaving’?”

From two pieces of evidence, we suspect so. Firstly, as one representative Diary noted

“But I learned the experience of how to conduct a science experiment. I searched the information for the problem, tried to think of alternative ideas to come up the good algorithm, and applied various algorithms to solve the puzzle”.

Secondly, in the examination script a student wrote in reply to a conceptually difficult question

“A genetic algorithm is not suitable (for this problem) because after research with the ‘Sudoku’ assignment a genetic algorithm will unlikely find the exact solution for this problem and it will find a ‘close enough’ answer.”

To better answer the question above regarding learning outcomes, a more detailed analysis of the submitted Research Diaries is planned using an empirical research methodology known as phenomenography. Previous phenomenographic research (2002Akerlind, 2005; Booth & Ingerman,) has been applied successfully in the examination of student learning and the ways in which it is experienced. In our instance, we are encouraged by the observation made by Berglund (2005) that “Phenomenography has proven helpful in computer science education since it allows a researcher to focus on computer science concepts and principles, while studying the learning of these concepts from the students’ perspective”. A more detailed examination using this approach will illuminate how students were constructing their internal knowledge whilst completing the Sudoku assessment.

Acknowledgement

The authors wish to thank the anonymous reviewers for their encouragement, insights and constructive criticism that helped shape the final version of this paper.

References

- Akerlind, G. (2005). Phenomenographic methods: A case illustration. In J. Bowden & P. Green (Eds.), *Doing developmental phenomenography* (pp. 103-127). RMIT University Press.
- Abraham, M. R. (1982). A descriptive instrument for use in investigating science laboratories. *Journal of Research in Science Teaching*, 19 (2), 155-165.
- Becher T. & Trowler, P. (2001). *Academic tribes and territories: Intellectual enquiry and the culture of disciplines* (2nd ed.). The Society for Research into Higher Education & Open University Press.
- Berglund, A. (2005). *Learning computer systems in a distributed project course: The what, why, how and where*. Uppsala Dissertations from the Faculty of Science and Technology. 62. Acta Universitatis Upsaliensis.
- Biggs, J. B. (2003). *Teaching for quality learning at university* (2nd ed.). The Society for Research into Higher Education and Open University Press. Great Britain.
- Biggs, J. B. & Collis, K.F. (1982). *Evaluating the quality of learning: The SOLO taxonomy*. New York: Academic Press.
- Booth, S., & Ingerman, A. (2002). Making sense of physics in the first year of study. *Learning and Instruction*, 12(5), 493-507.
- Borg, S. (2001). The research journal: A tool for promoting and understanding researcher development. *Language Teaching Research*, 5(2), 156-177.
- Chalmers, A. F. (2003). *What is this thing called science?* (3rd ed.). Queensland, Australia: University of Queensland Press.
- Epstein, S. L. (1999). Game Playing: The Next Moves. *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference* Orlando, Florida, United States, 987 – 993. American Association for Artificial Intelligence, Menlo Park, CA, U.S.A.
- Feldgen, M. & Clua, O. (2004). Games as a motivation for freshman students to learn programming. 34th *ASEE/IEEE Frontiers in Education Conference*. October 20-23, 2004, Savannah, GA. USA.
- Giguette, R. (2003). Pre-games: Games designed to introduce CS1 and CS2 programming assignments. *Proceedings of SIGCSE Conference 2003*. Feb 19-23. Reno, Nevada. USA
- Greening, T. (2000). Students seen flocking in programming assignments. *Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and technology in Computer Science education*. Helsinki, Finland. 93 – 96.

Thinking and Behaving Scientifically in Computer Science

- Jenkins, T. (2001). Teaching programming – A journey from teacher to motivator. *The 2nd Annual Conference of the LSTN Center for Information and Computer Science*.
- Jones, J. S. (1987). Participatory teaching methods in computer science. *Proceedings of the eighteenth SIGCSE technical symposium on Computer science education*, 155-160. St. Louis, Missouri, U.S.A.
- Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice Hall.
- Kolb, D. A. & Fry, R. (1975). Towards an applied theory of experiential learning. In C. L. Cooper (Ed.), *Theories of Group Process*. Chichester: John Wiley.
- Knowles, M., Holton, E. & Swanson, R. (2001). *The adult learner: The definitive classic in adult education and human resource development* (5th ed.). Woburn, MA: Butterworth-Heinemann.
- Lorsbach, A & Tobin, K. (1992). Constructivism as a referent for science teaching. *Research Matters to the Science Teacher*, National Association for Research in Science Teaching Monograph no, 5, 7.
- McDowell, L. (1995). The impact of innovative assessment on student learning. *Innovations in Education and Training International*, 32(4), 302-313.
- Miller, C. M. L. & Parlett, M. (1974) *Up to the mark: A study of the examination game*. London: Society for Research into Higher Education.
- National Academy of Sciences. (1995). *National Science Education Standards*. Chapter 1. Retrieved November 19, 2005, from <http://www.nap.edu/readingroom/books/nses/html/1.html#goals>
- Phillips, E. M & Pugh, D. S. (2000). *How to get a PhD: A handbook for students and their supervisors* (3rd ed.). Open University Press.
- Popper, K. (1972). *The logic of scientific discovery* (3rd ed.). London: Hutchinson.
- Ross, J. M. (2002). Guiding students through programming puzzles: Value and examples of Java game assignments. *SIGCSE Bulletin*, 34(4), 94 –98.
- Stewart, N. F. (1995). Science and computer science. *ACM Computing Surveys*, 27(1).
- Sudoku. (2005). Sudoku web site. Retrieved November 19, 2005, from <http://www.sudoku.com/>
- Snyder, B. R. (1971). *The hidden curriculum*. New York. USA: Knopf.
- Upchurch, R. L. & Sims-Knight, J. E. (1997). Integrating software process in computer science curriculum. *Proceedings of 1997 ASEE/IEEE Frontiers in Education Conference*. Pittsburgh, PA, U.S.A.
- Ylijoki, O-H (2000). Disciplinary cultures and the moral order of studying – A case study of four Finnish university departments. *Higher Education*. 39, 339-362.

Biographies



Anne Venables is a lecturer in Computer Science at Victoria University, Melbourne, Australia. She has research and teaching interests in artificial intelligence and intelligence systems. As a former secondary Science and Mathematics teacher who has migrated into tertiary education, Anne is also interested in innovations in education and has previously published in this field.



Grace Tan is a senior lecturer in Computer Science at Victoria University, Melbourne, Australia. Her research interests include investigations of innovative teaching methods, the development of graduate attributes, and issues related to female students in computing courses and Grace has published in these areas.