Assessing University Students' Coping Behaviours While Learning to Program: Report on a First-Step Analysis

Neville Meyers
Melanie Fleming

Queensland University of Technology
e-mail: n.meyers@qut.edu.au
e-mail: m.fleming@qut.edu.au

Abstract

Being able to program is considered an essential outcome of students' computer science education. However, programming is almost universally regarded as a difficult part of the curriculum for many students, resulting in high failure rates. Therefore, the current research being conducted as the result of a teaching and learning grant at UniOZ aims to assess students' coping levels when learning to program. This research-in-progress paper reports on pedagogical issues and design considerations leading to the production of a pilot questionnaire. Included are preliminary descriptive statistics exploring gender differences in perceived abilities to cope with learning to program and indicating relevance of the questionnaire items. The final questionnaire to be presented at the Conference will expand on research directions to assess individual students' learning problems, and invite comment

Keywords

Computer programming - education; self-efficacy – computers

INTRODUCTION

The history of computers in education has been marked by what Molna (1997) has called a "confluence of changes" in both curriculum focus and content. Teaching and learning approaches have been varied, ranging from applications of cognitive science, to instructional models (Lemut et al. 1992), various curricula reforms to cope with new instructional demands (Dadashzadeh et al. 2002), other micro approaches such as a personalised system of instruction for teaching Java script (Emurian and Durham, 2002), to the phenomenological analysis of how (in programming contexts) learners 'learn' (Booth 1992).

In recent times there has also been greater curriculum focus on the teaching of computer programming (e.g., in particular, Java) according to the particular demands of electronic commerce and the networked economy (Reilly and Resilly, 2002) as well as projected "skills shortages" of programmers. Faced with international competition and growing skills bases in such countries as India and China, there has been concern that about the "death" of the American programmer if educational systems fail to cope with demands (Mander 2001). Despite these various approaches, all institutions are faced with one universal problem: how do learners "learn" in particular curriculum areas; moreover, where and how might learning problems be both identified and rectified? In part to answer this question, (Huba and Freed, 2000) have argued for a shift in focus from teaching to learning. These writers argue:

> New information about how people learn…is forcing a paradigm shift in the education community. Even faculty who have formal educational backgrounds in pedagogy are finding that major changes in their approach to teaching are needed. This is because researchers have discovered that students learn by constructing knowledge rather than by receiving knowledge from others (Huba and Freed 2000, p.xvi).

The present project thus builds on and extends earlier research by Bruce and McMahon (2002), Bruce et al. (in submission) and by Truong (2002) as well as other research identified in the literature review. As noted by Bruce and McMahon (2002, citing other research by Taylor et al. 2001) failure rates for programming courses "are often in excess of 40%" (p.3). Moreover, programming courses have for some time been universally regarded as intrinsically difficult (Carter and Jenkins 1999). Accordingly, further research into teaching and learning of programming (in particular, students' expectancies regarding perceived levels of difficulty within programming units, their motivations to succeed, as well as their efficacy levels to persist and cope with actual course content) is highly desirable.

Context

The basic rationale for the programming unit serving as the basis for this study is that all Information Technology professionals need a fundamental knowledge of programming and an understanding of the processes and issues involved in the software development life cycle. Although not all Information Technology professionals are programmers, all Information Technology professionals will be required to work with programmers at some time in their careers. Thus, graduates of Information Technology courses are expected to be able to develop the competency to solve “real world” problems (see for example Jackling et al. 1990).To fulfil these aims, support material is made available through the Online Teaching site. This material includes lecture notes, sample source code, tutorial exercises, solutions to tutorials, and past examination papers. However given that students enrolling in Information Technology courses have a diverse range of technological skills, pedagogical approaches that are able to cater to this diversity are required. In addition, the Environment for Learning to Program (ELP), a web-based instructional tool, provides students with hands-on programming experience as well as valuable feed-back on competency attainments during various stages of their semester. This adoption of the ELP commenced in Summer School 2002/2003, and is in use once more in first semester 2003. However, there has been no systematic or formal evaluation of its efficacy in helping learners “learn”.

Another aspect of the way that students approach their study of the programming unit needs to be emphasised: namely, learning transfer, given that students are expected not to have any previous grounding in programming, is extremely difficult. For many students, the “shock of the new” may be an intimidating experience posing extra challenges to students’ self-efficacy (i.e., confidence in their abilities to perform appropriate behaviours to master the skills required to be a successful programmer; see for example, Bandura 1997). In addition, students are to display “mastery learning” (Nightingale 1996) according to their tutorial activities and other assessment (i.e., assignments, class exercises with peer-review solutions, and personal appointments with tutors to trouble-shoot difficulties), all integrated with a problem-solving based text (Lambert and Osborne, 2002).

However, it is useful at this stage to outline the theory-driven nature of the research; as well, to expand on the theories that have been chosen to underpin the study.

Theoretical Support for the Study

The primary body of theory chosen to inform the study is social cognitive theory, in particular, self-efficacy theory (Bandura 1977, 1986, 1992,1997). Given the learner focus as outlined by Huba and Freed (2000), social cognitive theory allows a particularly useful focus to examine the beliefs and attitudes that students bring to this area of their study. This is because “cognitive processes play a prominent role in the acquisition and retention of new behaviour patterns” (Bandura 1992, p.79). Central to the adoption of this approach is Bandura’s (1977, 1986, and 1997) focus on self-efficacy – the extent to which individuals believe that they can perform appropriate behaviours to secure desired outcomes. Individuals may have latent abilities; whether they attempt to use them or not will also depend on a range of motivational influences, not least of all, on whether they construe that their environments – in this case, the learning environment they face when studying computer programming - will help or hinder their efforts (Bandura 1986. 1997).

Self-efficacy theory has been successfully applied in a wide variety of educational, work, and health settings. It combines principles of learning emphasising the human interactions between the individual and his or her environment as opposed to viewing such interactions as purely a function of the environment (see also the viewpoints expressed by B.F Skinner, the best-known exponent of the earlier ‘behaviourist’ tradition, as overviewed in (Bandura 1997). For a succinct contemporary overview of social-learning and social- cognitive theories, see Gerrig and Zimbardo, 2002, pp.448-454. According to social cognitive theory, for example (notably its primary architect (Bandura, 1977, 1986, 1997) the environment does determine behaviour, However, individuals also have expectancies and are so motivated to exercise choices (Kirsch 1999). They scan their social, work and other environments for likely opportunities and when so motivated, initiate behaviours to secure desired outcomes (Bandura 1986; Kirsch 1999). According to this theory an outcome expectancy is the individual’s assessment that, in a generalised sense, certain behaviours can lead to successful outcomes (Kirsch 1999). However, self-efficacy theory goes beyond mere expectancies: an efficacy expectation, as earlier indicated, is the belief that one can actually perform the specific behaviours to attain those same outcomes (Bandura, 1977). Accordingly, it is important, from the point of view of social cognitive theory, to view human (experiences and motivations as a ‘three-way’ interaction in which behaviour, cognitions, and environmental influences all have mutually interacting effects that can be empirically assessed (Wood and Bandura 1989b) p.362). Using factor analysis, regression analysis and other statistical procedures it is also possible to isolate predictive factors regarding which behaviours are most likely to achieve desired outcomes (Bandura 1987; Meyers 1999).

Self-efficacy as an important psychological construct has been used in a wide range of work, health, educational, and personal domains. For example, management self-efficacy has been found to be a useful predictor of coping abilities of managerial decision-makers in complex, uncertain, and critical aspects of organisational functioning (Wood and Bandura 1989a). Self-efficacy has also been extensively applied in various other organisational and work contexts as overviewed in (Meyers, 1999).Amongst health practitioners self-efficacy has been used to assess patients' coping behaviours across several therapeutic domains (see Maddux 1997 for overviews). In the field of education, Bandura and his colleagues have conducted experiments relating to the relationship of self-efficacy and grade goals to academic performance illustrating the causal power of self-efficacy as earlier reviewed (see Wood and Locke 1987). In these studies, self-efficacy has been found to influence level of performance, task choice, effort, persistence, thought patterns, and stress reactions (see evaluations in Bandura 1986, 1997). Of particular relevance to the present study, numerous applications of self-efficacy theory have been noted in relation to computer use where self-efficacy has been found to be predictive of the extent to which individuals will persevere and achieve desired outcomes (e.g., Compeau and Higgins 1995; Compeau et al. 1999; Hill et al. 1987; Kay 1993). Closer in specificity to the present study is Ramalingam and Wiedenbeck (1998) and their development and validation of scores on their computer self-efficacy scale to contribute to a better understanding of coping differences between male and female students when learning to program.

Finally, in the present study, self-efficacy theory can be operationalised to assess learners' confidence in their abilities to perform appropriate coping behaviours to achieve the learning outcomes they are seeking (essentially, to achieve their desired level of proficiency in computer programming). Thus, the self-efficacy construct is a useful base for developing measures of the learner's capacity to cope with taxing aspects of content in their programming subject and to highlight where – in which competency areas – problems are being experienced and need to be addressed.

Research Questions

The current research – the result of a teaching and learning grant within the IT Faculty of UniOZ - aims to develop a clearer understanding of the attitudes that students bring to their first programming course (consistent with the learner focus as outlined in Huba & Freed, 2000). Ancillary goals are to examine the effects of selected pedagogical approaches; in particular, the impact of web-based learning on students' perceived self-efficacy; as well as to identify at what stages remedial instruction for individual students might be necessary. The project seeks further to identify to what extent a programming subject challenges students' existing computing skills, knowledge and self-efficacy. Research questions include:

1. To what extent may a programming unit challenge students' general self-efficacy levels?
2. To what extent does a programming unit challenge students' general competency attitudes towards their use of computers?
3. To what extent do students perceive they can perform the appropriate coping behaviours in a domain-specific area such as learning to program in order to achieve successful learning outcomes?
4. To what extent might gender factors influence attitudes to programming and learning outcomes?
5. To what extent might the additional resources (specifically a web-based training program) help alleviate some of the stress associated with learning to program, and improve students' feelings that they can cope with the course content?

METHODOLOGY

The genesis of the study was in the researcher's interests to examine more closely whether the introduction of the web-based module the Environment for Learning to Program (ELP) was likely to achieve positive educational outcomes; that is, that the ELP would result in higher student pass rates and increased students' self-efficacy for the subject. Accordingly, semi-structured interviews were conducted with teaching staff (N=4) to ascertain their views. The ancillary goal was to isolate key learning issues, and in particular, the core coping behaviours that students were expected to acquire in order to pass the subject. Although an initial literature review had been carried out, a further literature review was conducted, following the interviews, to redefine the theoretical grounding of the research. Finally, in consultation with teaching staff, it was decided to develop a questionnaire addressing the following core areas:

Section 1: Background Details

Students' age, gender, student status, were sought. Crucial however in the interests of both learning transfer and previously-acquired competencies were the questions "Do You have any programming experience?" and if "yes" in what programming languages (C==, Pascal, Modula-2, and other). This is because sources of self-efficacy beliefs may be grounded in antecedent conditions (i.e., already-acquired learning) that might enhance students' sense of mastery and contribute to a more easy transition to their programming studies (see Bandura, 1992; Wood and Bandura, 1989b). As well, it will be important during the next stage of data collection to allow for gender differences given other research (overviewed in Wilson 2002) indicating that, in some studies, important differences have been found to exist between males and females in their experience with and attitudes toward computers (p.143).

Section 2: How Do Students Generally Feel About Computers?

Included were seven items taken verbatim from an earlier well-validated sub-scale developed by Kay (1993). Students were asked to rate on 7-point Likert scale ('strongly disagree' to 'strongly agree') their responses) to such statements as "I do not need an experienced person nearby when I use a computer", "I could probably teach myself most of the things I need to know about computers", "If I had a problem using the computer, I could solve it one way or another". The intent was to ask students to rate to what extent they might indicate reasonable levels of control with their general applications of computers (that is, could they usually make the computer do what they wanted it to do?). This would establish students' general comfort with using computers and allow cross-comparisons with data later to be obtained, more specifically, about the impact of learning to program on such levels of perceived control. In short, does learning to program to some extent neutralise students' earlier computer experiences and competencies; or, put another way, does learning to program pose extra – potentially, unmanageable - demands on students' self-efficacy?

Section 3: How Do Students Generally Feel About Their Problem-Solving Skills in Other Life Situations?

The purpose of this section was to ask students to reflect on their general perceived competence in dealing with life situations. For example, an ancillary area of research (Sherer, 1982; Sherer and Adams, 1983) has argued that self-efficacy need not be exclusively viewed (contrary to Bandura 1977, 1986, 1997) as a domain-specific set of competencies. Rather, these opposing theorists argue that a generalised self-efficacy scale (i.e., how one deals with life's stressors' generally) may also have its place alongside the situation-specific assessments. This line of thinking was intuitively appealing to the researchers for two reasons: firstly, general self-efficacy (GSE) might indeed contribute to one's sense of mastery in other domains; and secondly, GSE if included in the study might also allow assessments (through later statistical correlations) with the competencies that relate to programming efficacy in the questionnaire's Sections 4 and 5. In short, do the inherent difficulties in learning to program once more neutralize students' previously acquired competencies – their sense of mastery - in other life domains? To provide a framework with which to deal with this question, the revised (2000) version of the General Perceived Self-Efficacy Scale (Schwarzer and Jerusalem, 1995) was adopted. This scale has been validated and adopted in some dozen countries where it has typically yielded internal consistencies between alpha=.75 and .90. The 2000 version of the 10-item Scale is the most up-to-date (Personal Correspondence R. Schwarzer and N. Meyers, 15 March, 2003).

Section 4: How Do Students Rate Their Overall Programming Skills?

This was an 8-item section of the questionnaire asking students to rate their perceived skills in such domain-specific areas as "Read and understand a computer program written in Java" and "Write a computer program in Java" using a 7-point scale with ratings 'extremely low' to 'extremely high'. Items were devised in consultations with teaching staff. Along the gradations of specificity built into the questionnaire, they were designed to assess students' perceived confidence in key programming areas that might also be later correlated with their use of the Environment for Learning to Program (ELP) module.

Section 5: How Confident Are Students When They Use the ELP Specifically?

This was an extremely important focus and asked students to rate to what extent they were confident they might use the ELP as an instructional tool to achieve the 10 core competencies for which it was designed to assist them. Consistent with the recommended format for assessing self-efficacy, (Bandura 1986, 1997), students were asked to answer YES or NO to each of the ten competencies and if YES rank their level of competency from 1 (not at all confident) to 9 (totally confident). Included, representatively, were such statements as "I can easily use the ELP to get the information I need", "I can complete and run a program in Java when using the ELP", "I can get sufficient feedback on programming mistakes via the ELP", "I can manage my stress levels when working through topics on the ELP", amongst other items.

Section 6: What Were Students Final Expectancies About Continuing to Use the ELP?

Expectancies about perceived benefits and likely successful outcomes may have powerful motivating effects when individuals appraise situations as they consider courses of action (Kirsch, 1999). In the present study, such assessments are important. The survey will be administered twice during the semester to measure the differences in students' expectancies as they progress in the unit. (early in the semester when students have received some training in use of the ELP and later in the semester when the results of this training are more evident). These sets of measures will allow pre-test and post-test assessments across the five crucial areas in the questionnaire. Accordingly, students have been asked via a 10- point scale to rate "Highly unlikely" to "Highly Likely" the strength of their expectancies.

Pilot Questionnaire

A pilot questionnaire was developed using well-accepted design principles (Leedy and Ormrod, 2001). It was also checked for item clarity and consistency with teaching staff (n=5) responsible for instructional design of the unit. Finally, the pilot questionnaire was administered to a large tutorial group (n-35 student enrolled first-semester in the programming unit). Students were invited to comment on unclear wording and other inconsistencies, and a small number (N=3) offered practical suggestions.

Restrictions on length of this conference paper do not permit inclusion of the questionnaire here. However, it is intended to table the questionnaire during presentation of the authors' conference paper; copies of the questionnaire will also be distributed at the conference to interested persons for additional comment. For general interest, however, a brief interim report on the results of the pilot questionnaire are included in the next section.

PRELIMINARY SURVEY RESULTS

What follows is a brief descriptive analysis of the results of the questionnaire pilot. Because the results are only preliminary, no attempt has been made to generalise from (or establish linkage with) the research literature. Instead, these results are meant to be exploratory and illustrative only. In the next phase of the study, the questionnaire will be administered second semester 2003 to a larger, potentially more representative sample (N=400 students) and fully analysed using appropriate statistical techniques according to recommended procedures (Tilley 1996). Encouragingly at this point however, as the basis for this initial reporting, the questionnaire items have in almost all cases achieved a good range of responses so that some form of preliminary, descriptive analysis may be useful.

Selection criteria and exclusions

There were a total of 33 participants to the pilot questionnaire. Of these all of the female students responses were used (n=10). Ten male participants' responses were randomly selected for inclusion in the analysis. (The additional surveys will be used to expand on these results in a subsequent study.) This meant that a total of 20 survey responses have been included in this analysis. All of these participants were undergraduate students and did not form part of the postgraduate student cohort.

It can be seen from table 1, that female students were most commonly between the ages of 18 to 24 whilst male students were most commonly below the age of 18. Additionally, females and males had most commonly used computers for 6 to 10 years (Table 2). However, the distribution of this data appears to be positively skewed for males and negatively skewed for females. That is, males were more likely to have spent more years using computers than females (for example no females spent more than 11 years using computers whereas 3 males had).

Table 1: Age of participants

	<18	**18 to 24**	**25 to 31**
Females	3	5	2
Males	6	4	0

Table 2: Years using computers

	1 to 5	**6 to 10**	**11 to 15**	**16 to 20**
Females	4	6	0	0
Males	1	6	2	1

Moreover, the prior programming experience of the participants was also noted. For example, the difference in the number of participants with prior programming experience is not great, with 6 females having learnt languages other than Java and 8 males having learnt languages other than Java. However of the females with prior programming experience, none claimed knowledge of more than 1 other language. However all of the males (except for 1) had knowledge of 2 other programming languages. One male participant claimed to know 4 other languages in addition to Java and claimed that his knowledge of two of these languages was expert. Females rated their level of proficiency from foundation (the lowest level) to good, with foundation level proficiency being most frequently reported. Males rated their knowledge from foundation to expert, with fair and good proficiency levels being most frequently cited.

Table 3 presents the participants average scale response to each main section of the survey. Higher numbers as registered on the pilot questionnaire indicate greater levels of confidence in an individual's ability to deal with computers; cope with general life problems; rate their programming ability; indicate their ability to use an online learning package designed to assist them to learn to program, and suggest their future learning outcomes.

Table 3: Overall ratings for each section of the questionnaire

	Females	**Males**	**Total possible scale value**
How do you feel about using computers?	30.8	40	49
General problem solving	31.6	31.4	40
Rating programming ability	27.8	37.7	56
Confidence using ELP	45.9	68	90
Learning outcomes	32.8	36.1	60

In four of the scales making up the survey, females' responses were much lower than those of their male counterparts. They tended to feel less confident in their abilities to use computers, less confident in their programming abilities, less confident about using the online learning package and less certain that they would achieve the learning outcomes that they wanted to in the course. This is despite rating themselves very similarly in their general abilities to handle day-to-day problems (females rated their general problem solving capabilities at 31.6 on average, whilst males rated theirs at 31.4 on average).

DISCUSSION

Discussion can admittedly be only limited given the exploratory nature of the pilot study the main aim of which was to validate the questionnaire. Nonetheless, drawing together interim findings it has been found that:-

- some students who rate themselves as expert programmers in other languages do not feel confident in their ability to program in Java
- females appear to rank their confidence in using technology as much lower than males
- some students who rank themselves as generally very competent in their everyday lives, feel much less confident with computers and programming
- conversely some students who rank themselves as being very confident in using computers and in their programming skills, do not feel at all confident in their everyday lives
- students showed very varied rankings in their abilities to program in Java when compared to other programming languages with which they were already familiar

As mentioned, the above results are exploratory only. Although only preliminary, the above results do suggest that the pilot questionnaire is tapping into useful constructs – constructs that await further, and more rigorous, empirical analysis.

Consequently, further conclusions as well as additional linkages to the literature are left pending the next phase of the study where appropriate statistical procedures (factor analysis, regression analysis, amongst others) can be adopted (Tilley 1996).

CONCLUSION

Insufficient evaluation has occurred to date at UniOZ's Faculty of Information Technology regarding the specific attitudes that students bring to their programming instruction. Neither has adequate qualitative or quantitative data been obtained regarding a core intervention – the Environment for Learning to Program module – nor adequate consideration been given whether further instructional and financial resources are warranted in this area. In particular, the proposed expanded research will further address core competencies that students need across several areas of their instructional experience. To date, the preliminary analysis suggests that the pilot questionnaire once administered to a larger sample should yield useful data. Moreover, the project may indicate where further research might be needed to enhance understanding of the difficulties that students face while learning to program. .

FURTHER RESEARCH

Future research as part of the current teaching and learning grant will investigate links between students' perceptions of self-efficacy and their actual performance in their programming course. A prime focus of the on-going study concerns the extent to which self-perceptions of ability (i.e., self-efficacy beliefs), using pre-test and post-test measures, can be assessed after increased exposure to the instruction modules specifically developed for the programming unit. An exploration of students' withdrawal behaviours as they are confronted with learning difficulties when learning to program as well as the impact of such behaviours on failure rates are of additional interest. (Thompson and Dinnel 2003);

REFERENCES

Bandura A. (1992) Self-efficacy: Toward a Unifying Theory of Behavioral Change, In 2nd (Ed.), *Management and motivation* (pp. 78-89), Penguin Books, London.

Bandura A. (1997) *Self-efficacy: The Exercise of Control*, W.H. Freeman: New York.

Bandura A. (1986) *Social Foundations of Thought and Action: A Social Cognitive Theory*, Prentice Hall, Englewood Cliffs, NJ.

Bandura. (1977) *Social Learning Theory*, Prentice Hall, Englewood cliffs, NJ.

Booth, S. (1992) *Learning to program: A Phenomenographic Perspective*, Göteborg, Sweden, Acta Universitatis Gothoburgensis

Bruce C. and McMahon, C. (2002) *Contemporary Developments inTeaching and Learning Introductory Programming: Towards a Research Proposal*, QUT Faculty of Information Technology, Brisbane.

Bruce C. McMahon C., Buckingham L., Hynd J and Roggenkamp M. (In submission) *Ways of Experiencing the Act of Learning to Program*, QUT Faculty of Information Technology, Brisbane.

Carter, J and Jenkins T. (1999) *Gender and programming: What's going on?* Proceedings of the 1999 4th Annual SIGSCE/SIGCUE Conference on Innovation and Technology in Computer Science Education, ITiCSE' 99ACM SIGSCE/SIGCUE, 1-4.

Compeau D.R. and Higgins G.A (1995) *Computer self-efficacy: Development of a Measure and Initial Test*, MIS Quarterly, 19, 189-212.

.Compeau, D. R., Higgins, C. Huff, S. (1999) Social Cognitive Theory and Individual Reactions to Computing Technology: A Longitudinal Study, *MIS Quarterly*, 23(2), 145-158.

Compeau D. R and Higgins C. A. (1995) Computer Self-efficacy: Development of a Measure and Initial Test, *MIS Quarterly*, 19(2), 189-212.

Dadashzadeh M., Saber A. and Saber S. (Eds.). (2002) *Information Technology Education in the New Millennium*, IRN Press, Hershey, PA.

Emurian, H. H. and Durham, A. G. (2002) A Personalized System of Instruction for Teaching Java, In S. Saber (Ed.), I*nformation Technology Education in the New Millennium*, IRM Press, Hershey, PA.

Hill, T., Smith D. and Mann, M. F. (1987) Role of Efficacy Expectations in Predicting the Decision to Use Advanced Technologies: The Case of Computers, *Journal of Applied Psychology*, 72(2), 307-313.

Huba, M. E. and Freed, J. E. (2000) *Learner-centred Assessment on College Campuses*, Alyn and Bacon, Boston.

Jackling, N., Lewis, J., Brandt, D. and Sell, R. (1990) Problem- solving in the Professions, *Higher Education Research and Development*, 9(2), 133-149.

Kay, R. H. (1993) A Practical Research Tool for Assessing Ability to Use Computers: The Computer Ability Survey, *Journal of Research on Technology in Education*, 26(1), 16-27.

Kirsch, I. (1999) *How Expectancies Shape Experience*, American Psychological Association, Washington, DC.

Lambert, K. A. and Osborne, M. (2002) *Java: A Framework for Programming and Problem-solving*, (2nd ed.) Brooks/Cole.

Langer, E. J., & Abelson, R. P. (1983) *The Psychology of Control*, Sage Publications, Beverly Hills, Calif.

Leedy, P. D. and Ormrod, J. E. (2001) *Practical Research: Planning and Design* (7th ed.), Merrill Prentice Hall, Upper Saddle River, N.J.

Lemut, E., Du Boulay, B. and Dettori, G. (Eds.). (1992) *Cognitive Models and Intelligent Learning Environments for Learning Programming*, Springer, Berlin, 1992.

Maddux, J. E. (Ed.). (1997) *Self-efficacy, Adaption, and Adjustment: Theory, Research, and Application*, Plenum: New York.

Mander, K. (2001) The Decline and Fall of the American Programmer? *Communications of the ACM*, 44(7), 43-44.

Meyers, N. T. (1999) *Personal Control Beliefs and SustainableTelecommuting: An Exploration of Relationships*, Unpublished Doctoral dissertation, Queensland University of Technology, Brisbane, Q'ld.

Nightingale, P. (1996) *Assessing Learning in Universities*, UNSW Press: Sydney.

Ramalingam, V. and Wiedenbeck, S. (1998) Development and Validation of Scores on a Computer Programming Self-efficacy Scale and Group Analyses of Novice Programmers, *Journal of Educational Computing Research*, 19, 367-381.

Reilly D. and Resilly M. (2002) *Java (TM) Network Programming and Distributed Computing*, Addison Wesley: New York.

Schwarzer R and Jerusalem M. (1995) The Generalized Self-Efficacy Scale, In M. Johnston (Ed.), *Measures in Health Psychology* (pp. 35-37), Nelson: London.

Sherer M. (1982) The Self-efficacy Scale: Construction and Validation, *Psychological Reports*, 51, 663-671.

Thompson T and Dale L.D. (2003). Construction and Initial Validation of the Self-worth Protection Scale, *British Journal of Educational Psychology*, 73, 89+

Tilley, A. J. (1996) *An Introduction to Psychological Research and Statistics* (3rd ed.), Pineapple Press: Brisbane.

Truong, N. (2002) *ELP: A Web Environment for Learning to Program*, Paper presented at the FIT Teaching and Learning Group.

Wood, R. and Bandura A. (1989a) Impact of Conceptions of Ability in Self-regulatory Mechanisms and Complex Decision-making, *Journal of Personality and Social Psychology*, 56, 407-415.

Wood, R. and Bandura A. (1989b) Social Cognitive Theory of Organizational Management, *Academy of Management Review*, 14(3), 361-384.

Wood, R. and Locke, E. A. (1987) The Relationship of Self-efficacy and Grade Goals to Academic Performance, *Educational and Psychological Measurement*, 47, 1013-1024.

Wilson, B.C. (2002) A Study of Factors Promoting Success in Computer Science Including Gender Differences, *Computer Science Education*, 12 (1-2), 141-164.

COPYRIGHT