An innovative approach to teaching first year programming supported by learning style investigation

Lynne Fowler
Murdoch University, Rockingham, Australia
lynne@eng.murdoch.edu.au

Vivian Campbell
Murdoch University, Rockingham, Australia
viv@eng.murdoch.edu.au

Daniel McGill
Murdoch University, Rockingham, Australia
daniel@eng.murdoch.edu.au

Geoff Roy
Murdoch University, Rockingham, Australia
geoff@eng.murdoch.edu.au

Abstract: *Murdoch University, School of Engineering Science, has since its inception in 1995 been actively embracing new challenges to improve teaching and learning within its courses. One of the aims of the school is to prepare students and empower them for the lifelong learning process.*

Since 1999 the use of Learning Style Inventories to monitor and address student learning has been undertaken, with great student interest and involvement. The importance of understanding the learning process is acknowledged by the inclusion of these surveys in our Engineering 1st year Foundation Unit, a general-purpose unit completed by most engineering students and many non-engineering students. By raising awareness of learning styles students are in a stronger position to take positive control of over their learning.

Many students struggle with 1st year programming courses and understanding of basic concepts. This research uses ***P-Coder****, a CASE tool developed within the school, to aid program design and development in 1st year Java programming units. This innovative approach is being monitored closely by performing pre and post tests throughout the year. Initial results highlight correlations between student learning styles and success in the pre test therefore identifying groups of students who may require additional help. This is a work in progress and tests will be continued at the start and end of each semester.*

If our research can identify students with preferred learning styles as "at risk" then we are in a better position to tailor support material to aid their learning knowledge uptake.

Keywords: CASE tool, learning styles, software engineering

© 2003 Australasian Association
for Engineering Education

Introduction

Murdoch University, School of Engineering Science, have since its inception in 1995, given a high profile to the role of teaching and learning within its courses. A decision in the early to mid 90s was taken to embrace the new technology of the time. The use of the World Wide Web was in its early stages but it was decided to provide all our courses online. This impacted our attitudes to teaching and learning because at the time this medium was relatively untested. Hence, the initial and ongoing interest in learning initiatives evolved in the school, which has benefited both staff and students. Work was started early in 2000 (Fowler, Allen, Armarego, & Mackenzie, 2000) on using learning style inventories with our students and has rapidly advanced as their usefulness and success was discovered. The two inventories that were researched and chosen were Kolb (Kolb, 1984) and Felder (Soloman & Felder, 1999).

In 2000 the School first offered its own Foundation Unit (McGill, Fowler, & Allen, 2002), which was aimed at providing an innovative and flexible mechanism to assist our students in developing study skills (Rowland, 2001). Murdoch University has several Foundation Units, which are general purpose units, and all first year students must study one of these units. The School decided that this new Foundation unit, *Interactions of Society and Technology,* with its *broad* interdisciplinary nature provided the ideal forum for a component on 'understanding your learning styles' (Fowler, McGill, Armarego, & Allen, 2002). Located on the Rockingham Campus the unit primarily attracts students from the Schools of Engineering Science, Information Technology and Commerce.

This decision affirms the recognition within the School of the value of students' understanding their own learning styles whilst complementing the development of Graduate Attributes (Rowland, 2001) and the Foundation Unit proved to be the best mechanism for developing and highlighting these learning skills within our students. This therefore, supports one of the School's aims to empower our students in their university and life long learning requirements, whilst also providing a mechanism to enable our staff to reflect on their own teaching styles and adopt different strategies where appropriate. The Foundation Unit is valuable for all students but in particular for EngFocus students who have entered an Engineering degree via a bridging course and may not have all the expected prerequisite skills.

EngFocus is an innovative program that had its first run as a pilot study by the then School of Engineering through the December/January period of 2002/03. As a pilot study the program enrolled only 12 students from three local high schools. The program was scheduled for 5 weeks of full-time study with a break over the Christmas/New Year period. This program was designed to provide a bridging structure for non-TEE (Tertiary Education Entrance) students who had expressed interest in studying engineering within the School of Engineering Science. None of the selected students had pre-requisites for entry to University let alone into Engineering Science. Broadly, the students were Year 12 graduates either from a wholly school-assessed background or a strong VET (Vocation Education and Training) educational focus.

The essential idea behind EngFocus was to provide an avenue for non-TEE, VET students to take part in a discipline-specific bridging course that also introduced them to the generic academic skills needed to make a successful transition into University study. These academic skills were in addition to the discipline-specific skills needed to study in the Engineering

© 2003 Australasian Association
for Engineering Education

field. By covering both aspects of their orientation to University the students are exposed to a range of study activities that would be reflective of the learning environment they are likely to engage within the University. The commitment to the students was that they would have the opportunity to experience both the skills and requirements of study within the area of Engineering studies as well as the skills and expectations demanded of them by the University in their first semester of study. This exposure to the breadth of University study was paramount, as the students had not been prepared for academic study through their school experience and, as we did not want to set them up to fail, we had to provide them with appropriate academic experience so that the decision on whether they were going to pursue tertiary study could be made from an informed position.

Both retention issues and the level of learning in first year are of paramount importance for students to succeed. There have been dozens of studies that suggest new techniques as recommended solutions and there have been a few studies that consider the attributes of students that are predictors of success (Goold & Rimmer, 2000) and (Wilson & Shrock, 2001). At Murdoch we are looking to enable students to be flexible in order to achieve success in all environments and have analysed students' learning styles in order to raise student awareness of learning issues (Fowler, Armarego, & Allen, 2001a). We feel supporting first year students in this way is critical to their survival in the early stages of their chosen degree, when they are most at risk.

This research has specifically focused on the use of a software package to help students understand the important concepts required in a first year programming course. By investigating learning styles and relating it to their successes in using our CASE tool, P-Coder, we will be in a better position to aid the students learning process.

P-Coder Case Tool

P-Coder is a CASE tool developed within the School of Engineering Science, by the 4th author, and is aimed as a support tool to assist in the teaching of novice programmers (students taking their first or second units in programming/computing). It is intended for use in relatively small scale programming tasks. It is not intended to be a full-scale development environment, and it will not scale to complex programming tasks. It has not been designed to replace the use of one of the many IDEs that can be used for producing larger and more complex programs.

The teaching of basic programming skills and the underlying knowledge has been a relevant topic for many years. Early interest was sparked in the 1970s with the development of the structured approached, followed by attempts to devise programming languages to support, or compliment, these approaches (e.g. Pascal). Other approaches to programming were also developed at this time (e.g. declarative styles through functional and logic languages). While these have been able to demonstrate considerable strengths in supporting programming tasks, they have not come to dominate the world of software development. Procedural programming concepts have evolved through a range of languages and they provide varying levels of support for the programmer for the programming tasks. In more recent times the evolution of O-O technologies has provided new challenges for teaching.

There is an underlying need to understand the very basic computational processes (sequence, iteration, selection and recursion) no matter what programming language is being used. In modern teaching practice it seems essential that both procedural and O-O concepts are required elements. The challenge is to get the balance right, and, if possible, demonstrate that

© 2003 Australasian Association
for Engineering Education

these concepts form part of a continuum of knowledge that is required by the competent student.

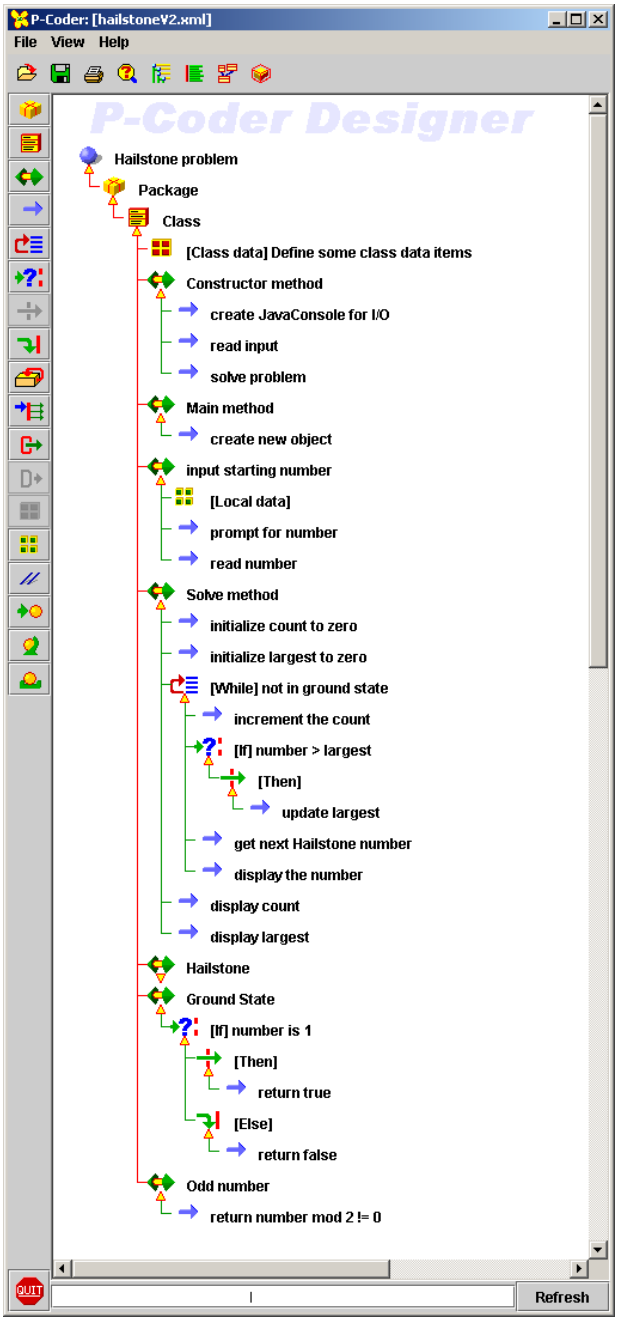

Figure 1: P-Coder Designer View

So, in the 21st century we are still faced with many of the same problems that students faced 20 or 30 years ago. These can be summarised as:

- difficulties in conceptualising the computational task and its solution starting from an informal description of the task,
- confusion between language syntax and the computational process,
- difficulties in devising and understanding the computational algorithm required for the task,

© 2003 Australasian Association
for Engineering Education

- lack of ability (skill, experience) to understand the flow of computation within a program,
- difficulties in using, and appreciating the advantages of, appropriate encapsulation and modularisation concepts,
- a general lack of understanding of O-O concepts in programming.

In essence we can summarise these by the fact that many novice programmers fail to appreciate the big picture while they struggle with the low level syntactical elements of the programming languages.

P-Coder is intended for use in the early stages of teaching programming skills. It has its origins in pseudocode principles, but also adds some additional O-O concepts that are integral to many modern programming languages. Pseudocode provides an intermediate step in the programming process – a step that can be seen to relate to both the informal specification and also to the final code. In its current form, P-Coder is Java oriented.

The P-Coder CASE tool enables students to design programs using pseudocode and, with additional specification, allows code to be automatically generated. It builds on the four key (and fundamental) computational building blocks (sequence, iteration, selection and recursion) with some added notation that provides other core (especially O-O based, and some Java) concepts to be presented within the framework. The emphasis for the students is now on design rather than syntax. Figure 1 illustrates the designer view within P-Coder, which is used as an entry point for the main programming constructs.

This semester is the first trial run of this innovative tool. After five weeks of the course student response is positive and staff perception is that understanding of important concepts appears to be better than in previous years.

Pre and Post Tests

Since P-Coder was developed at Murdoch it was considered important that an evaluation of the tool be carried out. Evaluation is important in order to clarify whether the technology enhances the student's understanding (Alstrum et al., 1996).

In order to assess the value and success of P-Coder and to monitor student learning, a multi-choice forty question online test has been developed. The test is a formative assessment since it is expected to assist students in the learning process. However, the main purpose of the test is to find out what students know, so it could be described as summative (Isaacs, 1994). The decision to create a test rather than use the standard assessment of assignments and exams allowed a quasi-experimental approach to be taken in this evaluation. Ethical considerations prevented the use of a control group.

This test includes questions covering the entire year's syllabus taught over two semesters. The same test is to be administered four times throughout the year namely week 2 semester 1, week 13 semester 1 (last week of semester), week 1 semester 2 and week 13 semester 2. The aims of the test are:

- to assess student learning,
- to motivate students to increase their score,
- to act as a pre and post assessment of the course.

© 2003 Australasian Association
for Engineering Education

The students are aware that the scores for the test are not used formally and are purely for self-assessment and research. However the first test has already shown increased student motivation, in that one student commented, "I have read the first five weeks of the course book ready for the test".

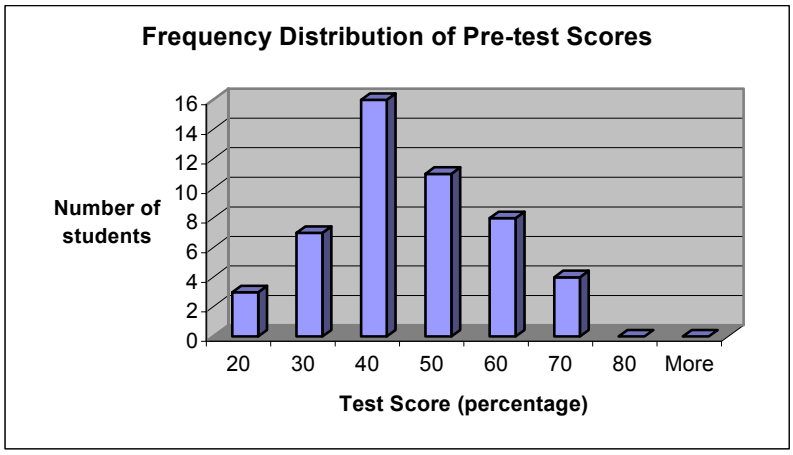

Figure 2: Frequency distribution of Pre-test scores

Results of the first administered test, week 2, semester 1(March 2003) were surprisingly good with an average of 40.8% and standard deviation of 12.6. However the test could not be done before the students had access to the computer labs and so was not carried out until week 2. In addition students were allowed to guess and no penalties were given for wrong answers. It has been shown that guessing is often of a greater benefit to able rather than less able students (Hutchinson, 1991). Although we are unable to prove this, a reduced standard deviation in later tests would be indicative. Figure 2 shows the distribution of scores, which will be compared to later tests.

Learning Style Inventories

Whilst there are numerous instruments for assessing learning styles, those advocated by Kolb, *Learning Style Inventory* (Kolb, 1984) and Soloman and Felder, *Index of Learning Styles* (Soloman & Felder, 1999) are well known, and accepted within education theory (Montgomery, 1995). Both instruments provide an efficient way of analysing our students' learning styles and complement each other on the information they supply.

The learning style inventories evaluate the way a person learns and how they deal with day-to-day situations in their life. Helping a person to understand how they learn and how individuals differ enables them to take positive control over their learning processes. It follows that individuals can be aware of and address the divergences between student and staff learning styles, and academic staff can use this awareness to develop material and teach in a greater variety of ways.

The importance of learning styles and their support on the construction of knowledge is therefore of paramount importance. The constructivist approach (Phillips, 1995) used in this research has focussed on the styles of learning that apply to either different categories of learners, or the learning of different categories of material, providing insights into individual

© 2003 Australasian Association
for Engineering Education

differences in learning and performance. The challenge is to identify the successful mental modelling strategies of the learner or to modify the learner's approaches to learning (McLoughlin, 1996).

Constructivist learning is described by Ernst von Glasersfeld's basic principles:

- that is knowledge is not passively received either through the senses or by way of communication, but is actively built up by the cognising subject,
- the function of cognition is adaptive and serves the subject's organization of the experiential world, not the discovery of an objective ontological reality.

(Heylighen, 1997)

Knowledge can be viewed as a constructed entity made by each learner, through a learning process and cannot be transmitted from person to person but needs to be constructed, possibly re-constructed, by each person. We acknowledge the two major views within the constructivist school of learning: cognitive oriented theories, stressing exploration and discovery, and socially oriented theories, emphasising collaboratory efforts of groups of learners.

Therefore looking at students learning styles to aid the learning process and the construction of knowledge is important. Also, raising students' awareness of issues surrounding their learning will lead to more effective learning practices and study outcomes.

Kolb Learning Styles Inventory

Kolb defines learning styles as one's preferred methods for perceiving and processing information (Jonassen & Grabowski, 1993). He views the learning process as a four-stage cycle: concrete experience (CE), feeling, followed by reflective observation (RO), watching, abstract conceptualization (AC), thinking, and active experimentation (AE), doing. CE and AC represent one continuum, how one prefers to perceive the environment or grasp experiences of the world. The second continuum, RO and AE represent how one prefers to process or transform information. By crossing the two continua, Kolb differentiates four types of learning: *divergers*, *assimilators*, *convergers* and *accommodators*.

The users' learning style, (Burns, 1989), can then be identified as either:

- Accommodator: *What if?* people. Often start with what they see and feel then plunge in and seek hidden possibilities. They learn by trial and error, and self-discovery,
- Diverger : *Why or why not?* These people study life as it is and reflect on it to seek meaning. They learn by being involved and need to listen and share with others,
- Converger: *How?* These people start with an idea and try it out, they like to find out how things work and learn by testing theories,
- Assimilator: *What?* people. These people come up with ideas and then reflect on them. They like to know what the experts think.

Our results build upon our previous studies (Fowler et al., 2001a), (Fowler, Armarego, & Allen, 2001b) and (Fowler et al., 2002) . The learning styles of our engineering students are diverse, and span all categories, (Table 1), indicating the variety of student types that our courses attract. This result is excellent given the multi-disciplinary nature of our curriculum content but we need to be able to cater for all students and their learning styles. Our staff show a greater tendency to be *assimilator* and *converger* types; this is in line with Kolb

© 2003 Australasian Association
for Engineering Education

(Kolb, 1984) stating that engineering is a good career area for *convergers* and that teaching suits *assimilators*.

Clients	No. of Clients	Accommodator	Diverger	Assimilator	Converger
Engineering 1st year Students	126	8%	18%	33%	41%
Engineering Staff	12	0%	17%	41.5%	41.5%
General Arts & Commerce 1st year Students	198	13%	13%	47%	27%
Year 12 all students	112	26%	10%	44%	20%
Computer Science, IT 1st year Students	66	5%	12%	56%	27%
G108 1st years 2003 only	**48**	**4%**	**8%**	**42%**	**46%**
4th year Engineering students	29	3%	7%	40%	52%

Table 1: Kolb *Learning Style Inventory* 1999 – 2003 cumulative results

Soloman and Felder Index of Learning Styles

The *Index of Learning Styles* (Soloman & Felder, 1999) is an instrument to assess learning preferences on four dimensions*; active/reflective*, *sensing/intuitive*, *visual/verbal*, and *sequential/global*. This instrument consists of forty-four simple questions each with a choice between two possible answers.

The results from Table 2 show the following mismatches between staff and students:

- in nearly all categories students are more *active* than *reflective* but our teachers are mainly *reflective.* The exception is Computer Science/IT and G108 programming students who are showing a more balance split,
- over 59 % of all students are *sensors*, yet our teachers tend to be *intuitive*,
- both staff and students show a heavy tendency to be *visual*, yet traditionally material is presented to them verbally or in written form,
- students show a slight tendency to be *sequential* learners but an increasing percentage are *global* learners, yet teaching is often narrowly focused.

Our results for students are similar to those of Mackenzie, (Mackenzie, 1998), who surveyed 75 Mechanical Engineering students.

© 2003 Australasian Association
for Engineering Education

The profile of the General Arts and Commerce students has been included for comparison in Table 1 and Table 2. The Kolb survey, (Table 1), has differentiated more clearly between the learning styles of these two groups. The greater tendency towards *assimilators* for the general arts students is consistent with Kolb's description of *assimilators*, as being less practical and more creative.

Clients	No of Clients	Processing	Perception	Input	Understanding
Engineering 1st year Students	126	Active 56%	Sensory 63%	Visual 77%	Sequential 56%
		Reflective 44%	Intuitive 37%	Verbal 23%	Global 44%
Engineering Staff	11	Active 27%	Sensory 36%	Visual 73%	Sequential 45%
		Reflective 73%	Intuitive 64%	Verbal 27%	Global 55%
General Arts and Commerce 1st year Students	200	Active 65%	Sensory 68%	Visual 76%	Sequential 54%
		Reflective 35%	Intuitive 32%	Verbal 24%	Global 46%
Year 12 all students	111	Active 59%	Sensory 59%	Visual 77%	Sequential 56%
		Reflective 41%	Intuitive 41%	Verbal 23%	Global 44%
Computer Science/IT 1st year Students	63	Active 49%	Sensory 70%	Visual 84%	Sequential 68%
		Reflective 51%	Intuitive 30%	Verbal 16%	Global 32%
G108 1st year 2003 only	**33**	**Active 48%**	**Sensory 63%**	**Visual 79%**	**Sequential 48%**
		Reflective 52%	**Intuitive 37%**	**Verbal 21%**	**Global 52%**
4th year Engineering students	29	Active 76%	Sensory 55%	Visual 86%	Sequential 59%
		Reflective 24%	Intuitive 45%	Verbal 14%	Global 41%

Table 2: Soloman and Felder *Index of Learning Style* Survey 1999-2003 cumulative results

© 2003 Australasian Association
for Engineering Education

Practical applications of our learning styles results are discussed in a previous paper (Fowler et al., 2001a). A suggestion (Felder, 1993) is to talk to students about their learning styles and the strengths and weaknesses associated with each style. We achieve this by incorporating a topic into our first year Foundation Unit to survey and discuss student learning styles.

A potential mismatch between the teaching styles of the staff and the learning style of students is highlighted in both Table 1 and Table 2. Students whose learning styles are compatible with the teaching style adopted within a course tend to retain information better, obtain better grades and maintain a greater interest in the course (Felder, 1993). Yet the diversity of learning styles in our students suggests that flexibility in teaching style is of considerable importance.

Analysis of Results

The relationship between the test results and preferred learning styles of our first-year students, in the G108 programming unit (using P-Coder), will be continued throughout the year as the tests are completed. Thomas et al (Thomas, Ratcliffe, Woodbury, & Jarman, 2002) in a similar study correlated assignment and exam results with Felder's learning style and showed that reflective students scored higher than active students and verbal students scored higher than visual students. This was in keeping with the notion (Felder, 1996) that engineering education is biased towards reflective, intuitive, verbal and sequential learners.

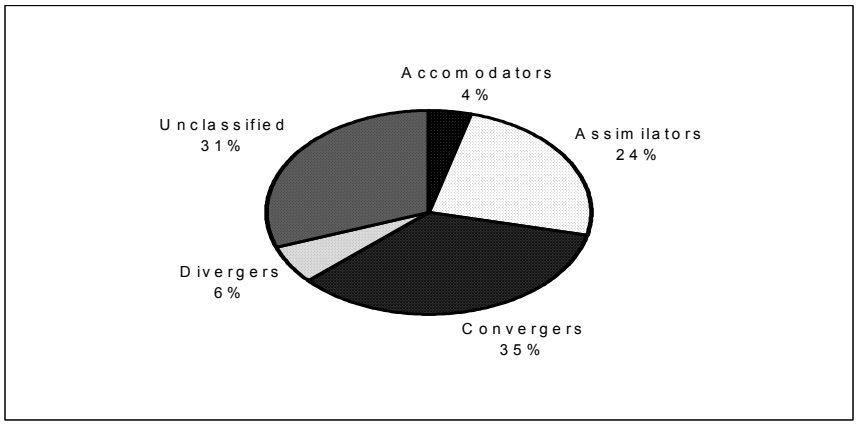

Figure 3: Kolb Learning Styles of 2003 Programming Students – Unit G108

Of the students that took the pre-test on programming, we have so far been able to trace the preferred learning style of three quarters of them, Figures 3 & 4. Of those that have been classified in our sample, *Convergers* and *Assimilators* are in far greater proportion than the very few *Accomodators* and *Divergers.* The scores were higher for intuitive learners, slightly higher for verbal and sequential learners whereas there was no differentiation between active/reflective learners. The correlation between learning style preferences and test scores will be investigated further as results become available.

The average scores of the four learning style groups are shown in Figure 5. Also shown is the mean for students for whom we have no preferred learning style information. While the *Accomodators* and *Divergers* were very small groups when compared to the other three their mean scores do appear to be significantly lower.

© 2003 Australasian Association
for Engineering Education

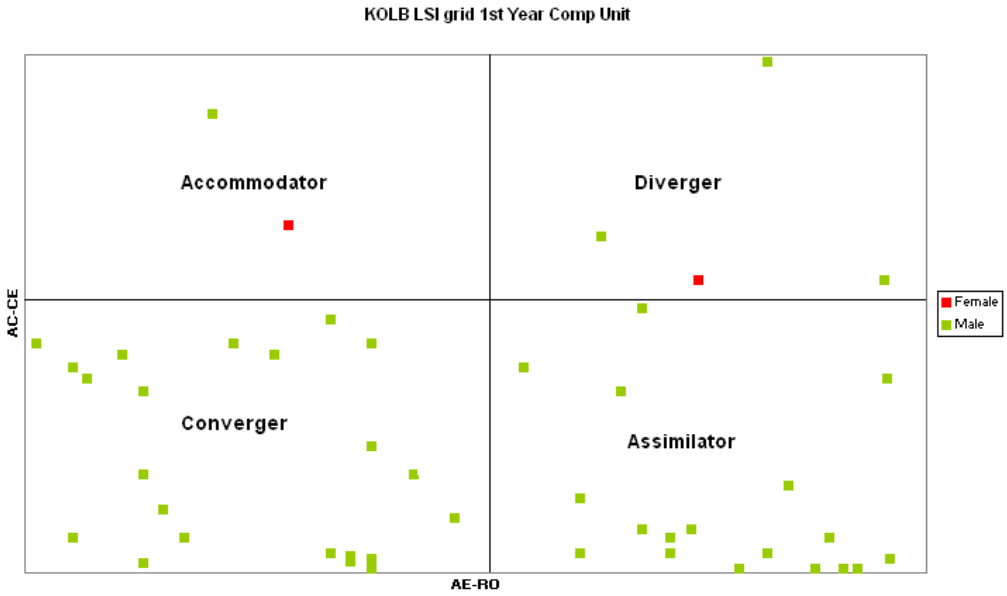

Figure 4: Kolb Learning Styles distribution for 2003 Programming Students – Unit G108/G109

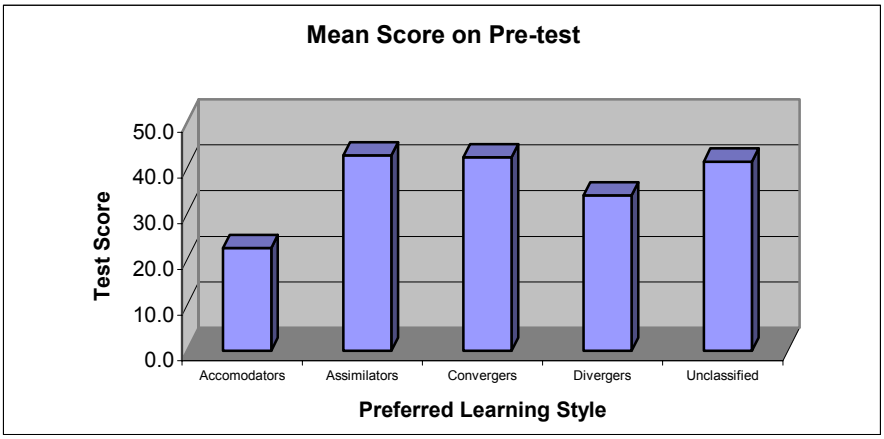

Figure 5: Mean Score for each of the Kolb Preferred Learning Styles

Conclusions

It has been shown in previous years that the *Accomodators* and *Divergers* have not been retained in our student cohort by fourth year. It may be that these students have failed because they are not able to learn in an environment where the teaching style favours *Assimilators* and *Convergers* or that they have modified their learning style to succeed. We feel by following this study through to completion by the end of this year we will have a better idea of which students succeed and therefore be in a better position to adjust our teaching styles to aid those who struggle with basic programming.

References

Alstrum, V., Dale, N., Bergland, A., Granger, M., Little, J. C., Miller, D. M., Petre, M., Schragger, P., & Springsteel, F. (1996). Evaluation: turning technology from toy to tool: report of the working group on evaluation.,, 201-217.

© 2003 Australasian Association for Engineering Education

Burns, S. (1989). There's More Than One Way to Learn. *Australia Wellbeing, 33*, 42-44.
Felder, R. (1993). Reaching the Second Tier: Learning and Teaching Styles in College Science Education. *Journal of College Science Teaching, 23*(5), 286 - 290.
Felder, R. M. (1996). Matters of Style. *ASEE Prism, 6*(4).
Fowler, L., Allen, M., Armarego, J., & Mackenzie, J. (2000). *Learning styles and CASE tools in Software Engineering.* Paper presented at the Flexible Futures in Tertiary Teaching. Proceedings of the 9th Annual Teaching Learning Forum,, Perth.
Fowler, L., Armarego, J., & Allen, M. (2001a). CASE Tools: Constructivism and its application to learning and usability of software of engineering tools. *Computer Science Education, 11*(3), 261-272.
Fowler, L., Armarego, J., & Allen, M. (2001b, November). *Learner Theory and its Application to Female Learner Support in Engineering.* Paper presented at the 10th International Women in Leadership Conference, Fremantle Western Australia.
Fowler, L., McGill, D., Armarego, J., & Allen, M. (2002). *Quantitative Learning Conversations: Constructivism and its application to learning in an engineering environment.* Paper presented at the HERDSA Conference, Edith Cowan University, Perth WA.
Goold, A., & Rimmer, R. (2000). Indicators of Performance in First-year Computing. *ACM SIGCSE Bulletin, 32*(2).
Heylighen, F. (1997). Epistemological Constructivism, *Principia Cybernetica Web*.
Hutchinson, T. P. (1991). *Ability, Partial Information, Guessing: Statistical Modelling Applied to Multiple-Choice Tests*.: Rumsby Scientific Publishing.
Isaacs, G. (1994). *Multiple Choice Testing*. Campbelltown, NSW, Australia: HERDSA.
Jonassen, D. H., & Grabowski, B. L. (1993). *handbook of Individual Differences Learning and Instruction*. London: Lawrence Erlbaum Associates.
Kolb, D. A. (1984). *Experiential Learning Experience as the Source of Learning and Development,*: Prentice-Hall.
Mackenzie, J. (1998). *Computer applications in chemical engineering.* Unpublished PhD, University of Canterbury, Canterbury, NZ.
McGill, D., Fowler, L., & Allen, M. (2002). *Flexible Learning and First Year Engineering Students.* Paper presented at the Australasian Association for Engineering Education, Canberra Australia.
McLoughlin, C. (1996). The implications of the research literature on learning styles for the design of instructional material. *Australian Journal of Educational Technology, 15*(3), 222-241.
Montgomery, S. M. (1995). *Addressing Diverse Learning Styles Through the use of Multimedia.* Paper presented at the Engineering Education for the 21st Century: Proceedings of the 25th Annual Frontiers in Education Conference.
Phillips, D. C. (1995). The Good, the Bad, and the Ugly: The Many Faces of Constructivism. *Educational Research, 24*(7), 5-12.
Rowland, F. (2001). *Foundation units and the graduate attributes: an audit*. Perth: Murdoch University.
Soloman, B., & Felder, R. (1999). *Index of Learning Styles (ILS),*. Available: http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSpage.html.
Thomas, L., Ratcliffe, M., Woodbury, J., & Jarman, E. (2002). Learning Styles and Performance in the Introductory Programming Sequence.
Wilson, B. C., & Shrock, S. (2001). Contributing to success in an introductory computer science course: a study of twelve factors. *Technical Symposium on Computer Science Education*, 184-188.

Acknowledgements

P-Coder is a CASE tool that has been developed by Professor Geoff Roy, Murdoch University, School of Engineering Science.