



Higher Education Research and Development Society of Australasia, Inc

Higher education in a changing world

Research and development in higher education

Proceedings of the

28th HERDSA Annual Conference

3-6 July 2005

Sydney, Australia

de Raadt, M., Hamilton, M., Lister, R. & Tutty, J. (2005) Approaches to learning in computer programming students and their effect on success, in *Higher education in a changing world, Proceedings of the 28th HERDSA Annual Conference, Sydney, 3-6 July 2005: pp 407.*

Published 2005 by the
Higher Education Research and Development Society of Australasia, Inc
PO Box 27, Milperra, NSW 2214, Australia
www.herdsa.org.au

ISSN: 0155 6223
ISBN: 0 908557 62 0

This research paper was reviewed using a double blind peer review process that meets DEEWR requirements. Two reviewers were appointed on the basis of their independence, expertise and experience and received the full paper devoid of the authors' names and institutions in order to ensure objectivity and anonymity. Where substantial differences existed between the two reviewers, a third reviewer was appointed. Papers were evaluated on the basis of originality, quality of academic merit, relevance to the conference theme and the standard of writing/presentation. Following review, this full paper was presented at the international conference.

Copyright© 2005 HERDSA and the authors. Apart from any fair dealing for the purposes of research or private study, criticism or review, as permitted under the Copyright, Design and Patent Act, 2005, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms and licenses issued by the copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers at the address above.

Approaches to learning in computer programming students and their effect on success

Michael de Raadt

University of Southern Queensland, Australia
deraadt@usq.edu.au

Margaret Hamilton

RMIT University, Australia
margaret.hamilton@rmit.edu.au

Raymond Lister

University of Technology Sydney, Australia
raymond@it.uts.edu.au

Jodi Tutty

Charles Darwin University, Australia
jodi.tutty@cdu.edu.au

Bob Baker

University of New South Wales, Australia
bob.baker@unsw.edu.au

Quintin Cutts

University of Glasgow, Glasgow, UK
quintin@dcs.gla.ac.uk

John Hamer

University of Auckland, New Zealand
J.Hamer@cs.auckland.ac.nz

Marian Petre

Open University, UK
m.petre@open.ac.uk

Simon

University of Newcastle, Australia
simon@newcastle.edu.au

Ilona Box

University of Technology, Sydney, Australia
ilona@it.uts.edu.au

Sally Fincher

University of Kent, UK
S.A.Fincher@kent.ac.uk

Patricia Haden

Otago Polytechnic, Dunedin, New Zealand
PHADEN@tekotago.ac.nz

Anthony Robins

University of Otago, Dunedin, New Zealand
anthony@cs.otago.ac.nz

Ken Sutton

Southern Institute of Technology, New Zealand
ken.sutton@sit.ac.nz

Denise Tolhurst

University of New South Wales, Sydney, Australia
d.tolhurst@unsw.edu.au

Abstract: *Within education research there has been sustained interest in developing models that can predict, or alternatively explain, student success. In computing education, attempts have been made to predict success in programming courses. Models previously used in this area have included a range of demographic, cognitive and social factors. These models emphasise presage factors. Biggs' 3P general model of student learning, by comparison, measures attitudinal factors. This multi-national, multi-institutional study investigates the effectiveness of an attitudinal measure, deep and surface approaches to learning (Biggs R-SPQ-2F questionnaire), to explain the success of students in introductory programming courses. This is then compared to both a cognitive and a demographic measure. The results indicate that across the eleven institutions in three countries the strongest correlation to success was found with the learning approach.*

Keywords: *learning approaches, introductory programming*

Introduction

Biggs (1978) describes a three stage model of learning: presage, process and product. Presage factors exist before the student enters the learning situation, and include such factors as prior knowledge, intelligence quotient, and home background. Process factors describe the learning context, which includes student perceptions. The product can be objective (eg exam marks) or subjective (eg satisfaction). Biggs (1987 pp. 70-71) cites cases where product was found to have been substantially influenced by process factors.

Within the discipline of computing success in learning to program has traditionally been attributed to presage factors. For example, participants undertaking the IBM Programmer's Aptitude Test (PAT) are asked to complete alphanumeric series, find matching figures and perform arithmetic reasoning. One trial of PAT included 63 American college students taking an introductory programming course (Mazlack, 1980). The performance of these students on PAT was correlated (Pearson) with their performance on the final exam for that course. While the correlation was significant ($p=0.038$), the correlation was only 0.23, and the author of that study concluded that PAT was not a reliable way of assessing an individual student's programming aptitude.

The drop-out and failure rates of computing degrees have been relatively high when compared to other university degree programs. These high rates have generated a sustained interest among computing academics for finding a predictor of "success" in programming where success is measured by a passing grade in an early programming subject. A recent paper in a prestigious computing journal described the "Grand Challenges" for Computing Education (McGettrick et al., 2005). One of the nominated challenges is the development of a "Programmer's Quotient" (PQ). The authors acknowledge the IQ test as their inspiration, describing such tests as measuring "*innate intellectual capability*". It is clear that McGettrick et al. place emphasis on presage factors.

In recent years, models to predict success in introductory programming courses have included demographic information, sometimes in conjunction with the results from cognitive tests. Demographic information has included gender, overall high school performance (eg In Australia the NSW UAI), family background (eg whether parents have a university degree), math background (eg American SAT math score), and previous programming experience. Recent models have also moved away from being predictive to being explanatory. Instead of only using factors known before a student commences the study of programming, explanatory models include factors describing the student experience as they study programming. A recent explanatory model used 12 factors in a linear regression model to predict student mid-term percentage mark in a programming course (Wilson, 2002). Of the twelve factors, the

most important factor was “comfort level”, a measure of student anxiety. Comfort level is a composite of seven indices, including the likelihood of asking questions in classroom, labs, and other circumstances.

All the afore-mentioned models of programming success place most emphasis on presage factors. This paper reports results from a study of programming success which included an instrument which measures how students learn, or approach learning. This inclusion broadens the range of factors considered. The instrument chosen was Biggs’ revised two factor study process questionnaire (R-SPQ-2F) which aims to measure students’ learning approaches to tasks in a particular teaching environment - the context-specific and situated nature of learning (Biggs, 1987; Biggs, 1999). The questionnaire yields two scores for deep and surface learning approach. Students adopting a surface approach build their view from facts and details of activities with the aim of reproducing material rather than making theoretical connections, while those adopting a deep learning approach seek to truly understand the material they are studying. While learning approach clearly interrelates to cognitive, demographic, and other factors that have been traditionally used in past models of success in programming, it is also sufficiently independent to offer the possibility that adding the Biggs instrument may significantly improve such models.

In this paper we examine the correlation between indicators from the Biggs instrument and the marks gained by students in their first programming subject. Students were also asked to perform a cognitive task, a paper folding test, and supply information on prior programming experience, for comparison with the data from the Biggs instrument. Using marks to measure success does open the question of how accurately these marks reflect the students’ programming ability. However as programming has no agreed, established ‘core’ list of essential programming concepts, let alone any robust multi-institutional instruments for assessing students’ acquisition of programming concepts, a student’s mark is the best performance indicator currently available.

Method

This section describes the instruments used in the study and how the background data was gathered. A kit containing the instruments was developed and trialled. Investigators from all the institutions were then trained prior to the data collection phase.

The Biggs instrument

Participants completed the revised questionnaire, consisting of 20 closed-response questions, scored on a 5-point Likert scale. Examples of questions include:

- Question 3: *My aim is to pass the course while doing as little work as possible.*
- Question 6: *I find most new topics interesting and often spend extra time trying to obtain more information about them.*
- Question 8: *I learn some things by rote, going over and over them until I know them by heart even if I do not understand them.*
- Question 13: *I work hard at my studies because I find the material interesting.*

Responses to questions are “*this item is...*”

- (Scoring 1 point) “*...never or only rarely true of me*”;
- (Scoring 2 points) “*...sometimes true of me*”;
- (Scoring 3 points) “*...true of me about half the time*”;
- (Scoring 4 points) “*...frequently true of me*”; or
- (Scoring 5 points) “*...always or almost always true of me*”.

There are four subscales calculated from the 20 questions: deep motive (DM), deep strategy (DS), shallow motive (SM) and shallow strategy (SS). For each of those subscales, five questions from the complete set of 20 contribute to the calculation of a score for that subscale. (The four example questions given above contribute to different subscales.) A score for deep approach (DA) is constructed by summing the DM and DS subscales. A score for shallow approach (SA) is constructed by summing SM and SS. For DA and SA, a participating student will score between 10 and 50; a score of 30 being neutral, lower scores being weaker, and higher scores being stronger in the approach.

Paper Folding

The *Paper Folding Test* (VZ-2) is taken from the ETS Kit of Referenced Tests for Cognitive Factors (Ekstrom, French, Harman, & Dermen, 1976). The test is designed to measure visualisation and spatial reasoning, based on the ability to manipulate and transform spatial patterns, and hence to recognise whether one image is a transformation of another. In this case, participants identify which pattern of holes would result in an unfolded sheet of paper after a hole is punched through an arrangement of folds.

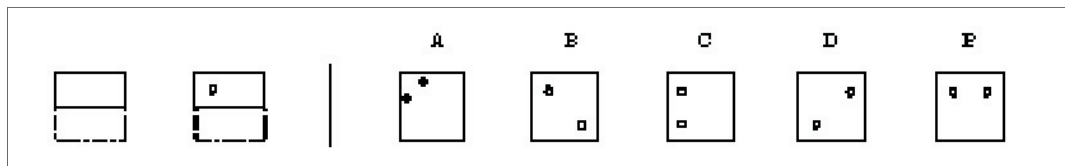


Figure 4. Example question from the Paper Folding Test

In Figure 4 the two illustrations left of the vertical bar represent a piece of paper being folded in half, followed by a hole being punched in the folded paper. To the right of the vertical bar, five options are presented for the pattern of holes that will be displayed after the paper is unfolded. The correct answer in this simple case is C. The test consists of 20 questions, in two sets of 10, with a time limit of three minutes per set. In the analysis of results below, the performance of students on this task will be a score out of 20, the total number of correct answers selected over the two sets of 10 questions.

The study described in this paper is not the first study to use a paper folding test as a means of assessing aptitude for computer programming. Evans and Simkin (1989) used paper folding in their study, but it only accounted for a small portion of their entire test.

Prior experience

Clearly, prior experience in most subjects, including computer programming, may be a significant aid to success at formal study. We asked students to nominate their prior level of experience in a set list of programming languages, on a scale of 1 (no experience) to 5 (extensive experience). Students provided this self-assessment early in the semester. Students could also suggest other programming languages and state their experience level.

Results

Results were collected from investigators in a central repository and analysed collectively (see Fincher *et al.*, 2005).

Participants

One hundred and seventy-seven volunteer participants were recruited from introductory programming courses at eleven tertiary education institutions in Australia, New Zealand and Scotland. Ages ranged

from 17 to 50 (three quarters were 22 or younger), with 137 males and 40 females. One institution contributed 32% of the total participants. The next highest contribution from a single institution was 8%. Of the 177 participants, 129 “completed” their first programming class. That is, only 129 students completed all assessment items associated with this one introductory programming course, (a reflection of the high drop out rates in the computing discipline, worldwide). The analysis of results in this paper focuses on these 129 students who completed.

Biggs questionnaire

The scores on the Biggs questionnaire of all students who completed are summarized in Table 1. In this table four questions which correlated highly with student mark are shown (these are the questions shown in the earlier description of the Biggs instrument). The twenty questions form groups of five DM, DS, SA and SM; SM and SS sum to form SA (Surface Approach); DM and DS sum to form DA (Deep Approach). DA and SA are the main two measures and these are highlighted. For each group or question, the range of possible values is shown in parentheses. The mean and standard deviation show that, as a population, the students are not strongly aligned with either deep or surface learning approaches, and their scores on the given questions are equivocal.

Group/Ques. (possible range)	DA (10-50)	DM (5-25)	DS (5-25)	SA (10-50)	SM (5-25)	SS (5-25)	Q3 (1-5)	Q6 (1-5)	Q8 (1-5)	Q13 (1-5)
Mean	28.8	14.5	14.3	23.5	10.8	12.7	2.3	2.8	2.2	3.1
Std. Dev.	7.0	4.0	3.7	7.2	3.7	4.0	1.3	1.1	1.2	1.1

Table 1. Aggregate statistics for complete students on the Biggs instrument

Paper Folding

On the paper folding test, participants in the current study scored a mean of 14.1 out of 20, with a standard deviation of 3.4. This result is slightly higher than, but consistent with, those of Ekstrom *et al* (1976) for 46 college students, with a mean score of 13.8 out of 20 and a standard deviation of 4.5.

Prior experience

Students’ responses to prior experience in programming are summarised in Table 2. Of all participants 77% claimed some past experience in at least one programming language. “Other” programming languages showed a higher average experience level than any set programming language. However, in this category, many students registered experience in technologies such as HTML, which are not general purpose programming languages.

	Java	C++	C	Scheme	Pascal	VB	Other
Mean	1.3	1.4	1.3	1.0	1.7	1.9	3.1
Std.dev.	0.7	0.9	0.8	0.1	1.2	1.3	1.4

Table 2. Student self-assessed prior experience from scale 1 (no experience) to 5 (extensive experience)

Correlations to mark

Pearson Correlations were calculated between student mark and the various data collected. The results are summarised in Table 3. All correlations shown are significant at the traditional 5% level.

Several elements of the Biggs questionnaire correlated better with mark than paper folding and prior programming experience. Prior experience with C and C++, however, does correlate strongly. Question 6 was “I find most new topics interesting and often spend extra time trying to obtain more information about them” and question 13 was “I work hard at my studies because I find the material interesting”; both are positive and correlated positively to mark.

Rank	Measure	Correlation	Significance	R ²
1	Biggs Instrument Question 6	0.31	0.001	0.10
2	Biggs Instrument Question 13	0.31	0.001	0.09
3	Prior Experience in C	0.30	0.0004	0.09
4	Deep Approach (DA)	0.29	0.003	0.08
5	Deep Motive (DM)	0.26	0.007	0.07
6	Deep Strategy (DS)	0.26	0.007	0.07
7	Surface Approach (SA)	-0.25	0.009	0.06
8	Biggs Instrument Question 8	-0.25	0.008	0.06
9	Surface Motive (SM)	-0.24	0.013	0.06
10	Prior experience in C++	0.24	0.007	0.06
11	Biggs Instrument Question 14	0.24	0.014	0.06
12	Surface Strategy (SS)	-0.23	0.020	0.05
...				
22	Paper Folding Score (out of 20)	0.17	0.047	0.03

Table 3. Correlations of factors with students' final mark

Significant correlations were found between mark and learning approach. Figure 5 shows a positive trend between student mark and a deep approach and a negative trend between mark and a surface approach. The lines in those figures are standard least squares lines of best fit. Figure 5 demonstrates that a linear regression model using Biggs DA or SA alone is not sufficient to accurately predict a student's programming mark. However, the trend lines suggest that the accuracy of traditional regression models which emphasise presage factors might be improved upon if those models were augmented to include Biggs DA and SA scores.

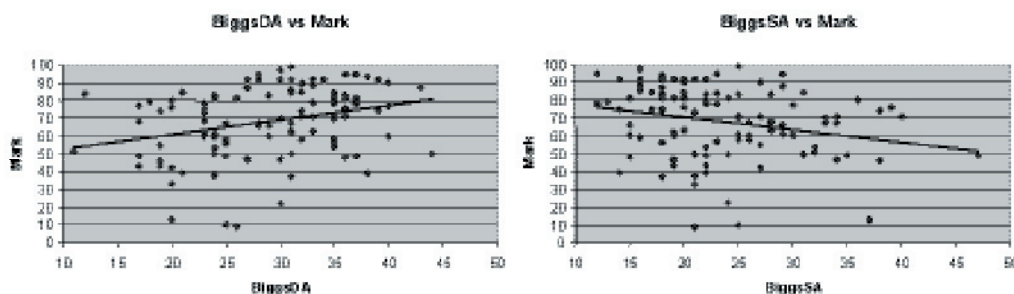


Figure 5. Deep approach against mark and surface approach against mark showing trendlines

Institutional differences

Of the instruments used in this study, only Biggs is context dependent. Students' adoption of their learning approach is interwoven with the environment. In a study such as this across multiple institutions with different curricula and assessments, the learning approach scores are a measure of the different learning environments as well as the individual student approaches. Some institutional variance is to be expected and Table 4 shows this to be the case. Each row of that Table shows, for a single participating institution, the correlation to student marks for DA and SA. For several of the institutions, the amount of data (n) is very small and no strong conclusions can be drawn. However, the correlations are broadly consistent, with the mark/DA correlation usually positive and the mark/SA correlation usually negative.

Institution	n	DA	SA
A	12	0.63	-0.58
E	13	0.42	-0.58
F	8	0.37	-0.02
H	13	0.43	-0.52
I	39	0.36	-0.29
J	12	0.54	-0.06
K	8	-0.48	-0.56
N	7	0.11	0.14
P	14	0.08	-0.22

Table 4. Correlation to student marks for DA and SA, by institution.

Discussion

As Table 3 shows, the correlation between learning approach and mark was significant and stronger than the correlations with most of the other factors investigated in this study. Only the correlation between previous C or C++ programming experience and mark was as strong. While previous programming experience contributing to better marks is logical this was not found to happen in all cases and somewhat surprisingly not for all programming languages. The small sample sizes may be part of the explanation. Another explanation is that C and C++ are frequently taught as the first programming language, as is Java, which has many syntactic similarities to C and C++. The last factor investigated in this study, the cognitive-based paper folding test, correlated poorly with final result in comparison to the other factors.

While these results indicate the potential of learning approach measures to help explain the success of students in introductory programming courses, some care is needed in interpretation. As discussed above, unlike cognitive and demographic factors, learning approaches are context specific and variable. While individuals do have a preference for a deep or surface learning approach that is relatively stable over time (Biggs 1987), it is not a fixed trait of the individual and can fluctuate over time and between tasks (Coffield, Mosely, Hall & Ecclestone, 2004).

Conclusions

The stereotype of the successful computing student is that of the 'geek' or 'nerd', a person with an unusual innate talent who spends hours alone with a computer. The results in this paper indicate that in computing, like other disciplines, learning approaches are a powerful determinant of success. There are

many reasons why students pass or fail particular subjects. It may be that the concept of a Programmer's Quotient is misdirected, since (just as with IQ) innate cognitive ability is but one factor of many that affect student grades. Even if there never is a reliable predictor of student success in computer programming, it may be worthwhile to show and discuss results like these to students and teachers alike, to discourage the attribution of success in programming to innate factors, and to encourage a more productive approach to learning.

In this paper, we examined the correlation between elements of the Biggs questionnaire and students' final marks in an introductory computer programming course. The questionnaire proved more strongly correlated with mark than did performance on a cognitive task (paper folding). Also, the correlations from several elements of the questionnaire proved to be stronger than correlations between mark and prior experience. These findings suggest that models for explaining student success in programming should be augmented to include data from the Biggs questionnaire, or a similar instrument.

References

- Biggs, J. (1978). Individual and Group Differences Study Processes. *British Journal of Educational Psychology*, 48, 266-79.
- Biggs, J. (1987). *Student approaches to learning and studying*. Australian Council for Educational Research.
- Biggs, J. (1999). *Teaching for quality learning at university: what the student does*. Open University Press.
- Biggs, J, Kember, D, and Leung, D. (2001). The Revised two-factor Study Process Questionnaire: R-SPQ-2F, *British Journal of Educational Psychology*, 71, 133-149.
- Ekstrom, R.B, French J.W and Harman, H.H. (1976). Cognitive Factors: Their Identification and Replication *Multivariate Behavioral Research Monographs*, 79(2).
- Evans, G and Simkin, M. (1989). What Best Predicts Computer Proficiency? *Communications of the ACM*. 32(11), 1322-1327.
- Fincher, S, Baker, B, Box, I, Cutts, Q, de Raadt, M, Haden, P, Hamer, J, Hamilton, M, Lister, R, Petre, M, Robins, A, Simon, Sutton, K, & Tolhurst, D. (2005). *Programmed to succeed?: A multi-national, multi-institutional study of introductory programming courses*. Technical Report 1/05, Computing Laboratory, University of Kent, Canterbury, Kent, UK. April 2005. Retrieved 26 May 2005 from: <http://www.cs.kent.ac.uk/pubs/2005/2157>
- Mazlack, L. (1980). Identifying potential to acquire programming skill. *Communications of the ACM*, 23(1), 14-17.
- McGettrick, A, Boyle, R, Ibbett, R, Lloyd, J, Lovegrove, L, and Mander, K. (2005). Grand Challenges in Computing: Education – A Summary. *The Computer Journal*, 48(1), 42-48.
- Wilson, B. (2002). A Study of Factors Promoting Success in Computer Science Including Gender Differences. *Computer Science Education*. 12(1-2), 141-164.

Acknowledgments

This study was supported by an ACM Special Interest Group in Computer Science Education (SIGCSE) Special Projects Grant, a grant from Computing Research and Education Association of Australasia (CORE), and a grant from the Computer Science Department of The University of Otago. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organisations.

Thanks to Caroline Wills for assistance with workshop organisation and transcription; to Diane Hagan, CSSE, Monash University, for her insight and assistance in conducting the study with Monash Students whilst a BRACE participant was on a Visiting Scholar Grant; to Warren Hill of Charles Darwin University in facilitating access to participants; to Charles Thevathayan of RMIT University and Dongmo Zhang of University of Western Sydney for facilitating access to participants, to Susan Snowdon for her contribution to the literature search; to Anthony Robins and Raymond Lister for exemplary workshop organisation and support; to Henk and Ilona Box Catering Inc for phenomenal food.

Copyright © 2005 Michael de Raadt, Margaret Hamilton, Raymond Lister, Jodi Tutty, Bob Baker, Ilona Box, Quintin Cutts, Sally Fincher, Patricia Haden, John Hamer, Marian Petre, Anthony Robins, Simon, Ken Sutton, Denise Tolhurst: The authors assign to HERDSA and educational non-profit institutions a nonexclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to HERDSA to publish this document in full on the World Wide Web (prime sites and mirrors) on CD-ROM and in printed form within the HERDSA 2005 conference proceedings. Any other usage is prohibited without the express permission of the authors.