

Predicting academic results in a modular computer programming course^{*}

Claudio Alvarez^{1,2}, Alyssa Wise³, Sebastian Altermatt¹, and Ignacio Aranguiz¹

¹ Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Santiago, Chile calvarez@uandes.cl

{[sjaltermatt](mailto:sjaltermatt@miuandes.cl), [iearanguiz](mailto:iearanguiz@miuandes.cl)}@miuandes.cl

² Centro de Investigación en Educación y Aprendizaje, Universidad de los Andes, Santiago, Chile

³ Steinhardt School of Culture, Education, and Human Development, New York University, NY, USA alyssa.wise@nyu.edu

Abstract. At present, computer programming skills are essential in engineering curricula and professional practice. In spite of this, and after decades of research in programming pedagogy, academic success in introductory programming courses continues to be a challenge for many students. In this research we explore the feasibility of predicting academic results in a modular computer programming course in a Chilean university (N=242), through measurement of psychometric variables linked to implicit theories of intelligence, error orientation, and students attitudes towards programming. Coincidentally with other recent studies conducted in Finland and Turkey, early measurement of implicit theories of intelligence did not emerge as a predictor of academic performance in the programming course. As for error orientation, students exhibiting mild measures of an error strain construct did seem to perform better than students with extreme measures. The variables with the highest predictive potential were found to be students' attitudes towards programming; namely, their perceived value of programming skills, and perception of programming self-efficacy. Substantial differences were noted in both latter constructs among male and female students. We discuss implications of our findings and future research prospects.

Keywords: Predictive Analytics · Computer Programming Course · Engineering Education · Psychometric Variables

1 Introduction

Programming skills are fundamental in virtually all branches of modern engineering practice [1]. Therefore, an introductory computer programming course is commonly taught early in engineering curricula, with the aim that students become proficient in solving engineering problems with computational tools, in specialty areas ranging from the most traditional, such as civil and mechanical engineering, to those fully engaged with Information and Communication

^{*} Research funded by CONICYT Fondecyt Initiation into Research grant 11160211.

Technologies (ICTs), such as computer science and software engineering. Due to their high relevance, ICTs in the training of engineers have been acknowledged as the latest among five significant shifts in engineering education during the last century [2].

In spite of the relevance of ICT-based tools in engineering education, high failure rates and attrition are common in programming courses in engineering schools. In countries such as Portugal and Brazil failure in introductory programming courses has been reported to be as high as sixty percent [3]. The possibility to anticipate which students could have greater learning difficulties when first acquainted with this discipline, or on the contrary, those who could be the most talented and successful, has prompted the interest of researchers and pedagogues for more than four decades. The research community still has no consensual response to this question.

In this research we explore the influence of various individual characteristics in students' academic achievement in a first programming course; namely, Implicit Theories of Intelligence (ITI) [4], error orientation [5, 6], and attitudes towards learning programming [7]. ITIs are known to influence students' achievement particularly in challenging and demanding academic situations. On the other hand, in learning a new complex skill such as programming, it is common for students to make mistakes frequently, and mistakes can be complex, combining several sources of error. Hypothetically, learner's behavioral response to errors can influence their learning ability and possibilities to succeed in a programming course. Lastly, students' attitudes towards learning computer programming can have an important role in shaping their learning experiences. We set out to explore the predictive potential of these constructs in academic performance, in the context of an introductory programming course for engineering freshmen in a Chilean university.

2 Factors influencing academic achievement in an introductory programming course

A fundamental research question of long standing is how best to predict a person's ability to master computer programming concepts. Attempts to resolve this question have persisted for more than four decades. Predicting success in an introductory programming course is difficult partly because of the lack of an established list of essential programming concepts, and the nonexistence of any robust instruments for assessing students programming proficiency [8]. Hence, most researchers in the field relate their findings from diagnostic instruments to the grades achieved by students in an introductory programming course. While this can be functional to research goals in each specific educational context, it hinders comparability among different studies, as well as the generalizability of research conclusions.

Studies before 1975 tended to explore learners' demographic background and past high school achievement. Between 1975 and 1981, prediction attempts were based on specific Programming Aptitude Tests (PATs), such as IBM's PAT [9].

Success with these performance measures was with mixed results. Studies using linear regression models could not explain more than half of the variance, with reported R-square values between 11 and 40 percent [10].

Cronan et al. [11] concluded that high school GPA, college GPA, sex, admissions test score in math, music reading, video game playing, size of hometown, and and prior computing classes in college variables could be used to classify students into upper or lower performance groups with a high level of accuracy. Rountree and his colleagues [12] determined the most reliable predictor of success in a programming course was the grade that the student expected to achieve.

Watson et al. [13] presented an approach for predicting students performance in a programming course, based upon analyzing directly logged data describing various aspects of their ordinary programming behavior. The approach could explain 42.49% of the variance in coursework marks. In spite that these results are comparable to linear regression models formulated in the 1970's and 80's, based on programming aptitude tests and past school achievement, the convenience of the log-based approach is that it can be administered in a non-invasive fashion, through closely logging students' actions in the programming environment.

Since the late 2000's, researchers have studied the influence of implicit theories of intelligence in learning programming, inspired by Dweck's mindset research [14]. Recently, Kaijanaho and Tirronen [15] conducted a study with a sample of Finnish students, with measurements based on the standard mindset instrument by Dweck. The authors found no correlation between the students' mindsets and their course grades, thus concluding that the effects of mindset on the results of the course are very small. Tek et al. [16], administered measurements of generalized implicit theories and of specific domain in programming along with measurements of self-efficacy. They constructed a multiple regression model involving these four predictors. None of the predictors was significant and the model failed to explain more than 10% of the variance in course grades.

3 Educational Context

The present research was conducted in the Faculty of Engineering and Applied Sciences at Universidad de los Andes, Santiago, Chile. The freshmen cohort of 2018 enrolled 242 students, 78.1% male and 21.9% female. The Programming course is compulsory for all engineering curricula, is delivered in face-to-face format, and is taught for freshmen in the first semester of career study plan, with a duration of 15 weeks. The course has a modular structure, that is, it consists of four modules lasting three weeks each, and is organized in successive time blocks with the same duration. By allocating five time blocks in a semester it is possible for a student to fail a module in the semester, and in such condition pass the course without having to re-enroll in the following semester. A student who has failed to pass all four modules of the course in the first semester may resume the course in the following semester, starting in the latest module he/she has not passed. If in the second semester the student fails to pass the course, he/she starts again from scratch with the freshmen cohort of the following year.

Table 1. Programming course contents.

Module	Topics
1	Introduction to computational problems and algorithms. Algorithm representation and modeling with MIT Scratch [17].
2	Python: Conditional flow control (if). Python: Iterative flow control (while). Python: Functions.
3	Python: String manipulation. Python: Lists, nested lists and slices. Python: Looping over collections using indices and iterators (for). Python: File access.
4	Python: Dictionaries. Python: Numeric Python (NumPy) [18]. Python: Charts (Matplotlib package). Python: Introduction to recursive algorithms.

Table 1 summarizes the contents of course modules. Each course module comprises three weekly lectures, three weekly tutorial activities in the computer lab, two graded lab assignments, an intensive two-day homework assignment, and a final exam. The latter has 70% of the weight in the module average, while lecture attendance, graded lab assignments, and homework account for 7.5% weight each.

4 Method

4.1 Measurements

Three instruments were administered in the current study, all of which were available in English and had to be adapted to Spanish. This was accomplished by following a process encompassing three parallel translations by two professional translators and the main author of the current paper. The three versions of the translated items were discussed by the three translators, which resulted in a consensual Spanish version for each item. Next, each translated item was back-translated to English and those items which presented inconsistencies were revised in their Spanish form to better resemble their original meaning. After the instruments were administered, Exploratory Factor Analysis (EFA) was conducted for each, and internal consistency was computed for every scale. An R programming environment based on RStudio 1.1 and R 3.4.1 was utilized in all analyses.

Implicit Theories of Intelligence (ITI) The eight item Implicit Theories of Intelligence Scale [14] was administered to measure students theories of intelligence. The scale comprises four incremental (*INC*) and four entity (*ENT*) theory items and assesses general beliefs about the fixedness vs. malleability of intelligence.

Error Orientation (EOQ) The Error Orientation questionnaire by Rybowskiak et al. [5], was developed for measuring how individuals cope with and how they think about errors at work. The instrument comprises eight subscales measuring error competence, learning from errors, error risk taking, error strain, error anticipation, covering up errors, communication about errors and thinking about errors. Of these, only the subscales measuring error competence (*ERRC*), error strain (*ERRS*; i.e., being strained by making errors and therefore fearing the occurrence of errors or reacting to errors with high emotion), learning from errors (*ERRL*), error risk taking (*ERRI*), and thinking about errors (*ERRT*) were adapted to Spanish language and to a general learning context.

Computing Attitudes Survey (CAS) Dorn and Tew [7] published an empirical validation and application of the Computing Attitudes Survey (CAS), an extension of the Colorado Learning Attitudes about Science Survey [20]. The instrument measures novice to expert attitude shifts about the nature of knowledge and problem solving in computer science. The fifth version of the instrument comprises five factors: Problem Solving (Transfer), Problem Solving (Strategies), Real-World Connections and Problem Solving (Fixed Mindset).

Performance Score (PSCORE) The Performance Score (PSCORE) is a cumulative performance indicator that permits objectively comparing and analyzing students' progress in the modular course, as in a given time block several modules can be simultaneously taught. At the end of time block i , and for each student j , $PSCOREi_j$ is computed considering the *last examination grade* EG obtained by the student j in module k , i.e., $EG_{jk}, k = 1...5$, with $EG_{jk} = 0$ if student j has not yet studied module k , and $EG_{j5} = 1$ if the student has passed all modules (zero otherwise), intended as a score reward. Finally, $PSCOREi_j$ is computed as follows:

$$PSCOREi_j = \log \sum_{k=1}^5 (10^k * \frac{EG_{jk}}{7})$$

As the grading system is based on a continuous 1.0 to 7.0 scale, examination grades are divided by 7.0 in *PSCORE* calculation to transform them to the $[0, 1]$ interval. The powers of 10 that multiply this quotient correspond to each of the modules, and this along with the logarithm applied to the sum ensures that *PSCOREs* from students in different modules in a given time block do not overlap, e.g., we avoid scores from high achievers in module 2 overlapping with low performers in module 3. We chose to construct the indicator solely relying on module examination grades, as exams are summative assessments that account for 70% of the final grade of each module.

4.2 Administration

The ITI and EOQ instruments having a general orientation were administered in paper format during the first week of freshmen classes. The CAS instrument

was administered one week after the start of time block 2 (week 4). The number of responses collected per instrument was 185 for RITI, 185 for EOQ, and 176 for CAS.

5 Results

Implicit Theories of Intelligence (ITI) With EFA for the ITI instrument the factor structure of the original instrument was preserved, as all items had factor loadings above 0.4 with varimax rotation. High internal consistency was found for both *ENT* (Cronbach's $\alpha = 0.91$) and *INC* ($\alpha = 0.86$) factors. Items in the ITI instrument had 5-point Likert scoring, and scores for *INC* and *ENT* were computed as the average of their respective items. Out of 185 students, 11 were found to be entity theorists (i.e., $ENT \geq 4.0$ and $INC \leq 2.0$, on a 1-5 scale), while 104 were found to be incremental theorists (i.e., $INC \geq 4.0$ and $ENT \leq 2.0$). The remaining 70 students could not be classified in either group. Intercorrelations among *INC* and *ENT* variables and *PSCOREs* were found negligible and non-significant. However, correlation between *ENC* and *INC* was -0.841 , which is consistent with the hypothesized relation between these constructs. Kolmogorov-Smirnov tests indicated no gender differences in distributions for both *INC* and *ENT* constructs ($D = 0.089$, $p > 0.05$ and $D = 0.208$, $p > 0.05$, respectively). Finally, multiple regression models were built with each *PSCORE* as response variable, and both *INC* and *ENT* as predictors. The regression's R-squared values were close to zero in all cases, and regression coefficients were found non-significant.

Error Orientation Questionnaire (EOQ) With regard to the EOQ instrument, the EFA resulted in a factor structure in which *ERRS* (Cronbach's $\alpha = 0.80$), *ERRL* (Cronbach's $\alpha = 0.81$), and *ERRI* ($\alpha = 0.70$) constructs could be identified with the same items as in the original instrument. A fourth factor combining items from *ERRC* and *ERRT* was identified, however, with low internal consistency ($\alpha = 0.65$). As in the original instrument, scoring was computed as the item average for each construct. All items had 5-point Likert scales.

Correlations among *ERRS*, *ERRL*, *ERRI* and *PSCOREs* were found to be negligible, i.e., all very close to zero, and non-significant. However, we did observe a slight academic performance advantage in students belonging to the second quartile (Q2) of the *ERRS* distribution (see Fig. 1). This is apparent specially in time blocks 4 and 5. In time block 5, 26% of students in Q2 had passed the course, compared to 16% in Q1 and Q4, and 15% in Q3. However, Kruskal-Wallis tests did not yield any statistically significant difference in *PSCOREs* among the four *ERRS* quartiles. A Wilcoxon rank sum test indicated that difference in *ERRS* is a statistically significant ($W = 3640.5$, $p < 0.05$) between male ($M = 3.30$, $SD = 0.72$, $Median = 3.4$) and female ($M = 3.89$, $SD = 0.78$, $Median = 4.1$) students.

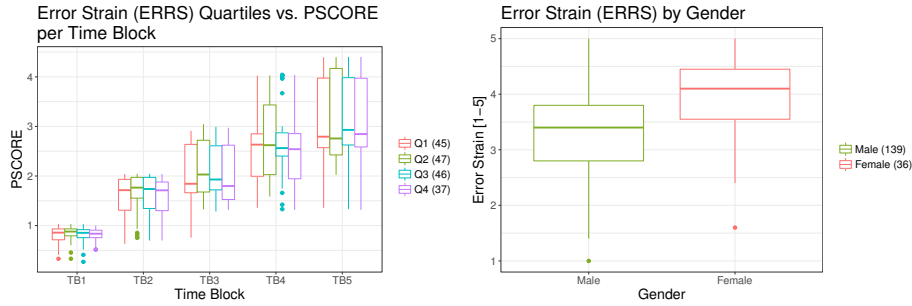


Fig. 1. PSCORE distributions *ERRS* quartile (left). *ERRS* by gender (right).

Computing Attitudes Survey (CAS) With the CAS instrument EFA was conducted utilizing polychoric correlations computed with WLSMV estimator [7]. The EFA with oblimin rotation yielded a factor structure different to the original instrument. Only 14 out of 25 items had factor loadings greater than 0.4, and these loaded into only three factors. The factors were interpreted by the current authors as Perceived Value (*PV*), i.e., students’ perceived value of course skills and knowledge, Perceived Self-Efficacy (*PSE*), and Endorsement of Ineffective Study Strategies (*EISS*) in learning programming. Internal consistency for the three factors was computed as ordinal alpha [19], which for *PV* was 0.86, *PSE* 0.87, and *EISS* 0.77. Statistically significant intercorrelations were found among *PSCOREs*, *PV* and *PSE*. Correlations among *PSCOREs* and *PSE* ranged between 0.28 (with *PSCORE5*) and 0.45 (with *PSCORE1*). In the case of *PV*, correlations ranged between 0.20 (with *PSCORE5*) and 0.30 with (with *PSCORE3*). In addition, statistically significant differences among male and female students were found in *PV* ($W = 2372, p < 0.001$) and *PSE* ($W = 2317.5, p < 0.001$) (see Figure 2).



Fig. 2. Perceived self-efficacy by gender (left). Perceived value by gender (right).

As the CAS instrument was administered after the conclusion of time block 1, we only considered students who passed module 1 in our analyses based on CAS, as the effects of failure in module 1 likely exert a negative bias on CAS measurements. Both *PSE* and *PV* distributions comprising the 140 students who passed module 1 and responded the CAS questionnaire were found highly asymmetric, with negative skewness (-1.29 and -1.36 , respectively) and high kurtosis (3.78 and 4.57 , respectively) (see Figure 2). Even though the distributions appear alike, correlation among *PSE* and *PV* is 0.44 , thus arguably these indicators contribute complementary information. In both distributions we labeled students scoring above percentile 50 as having High *PSE* and High *PV*, students in between percentiles 25 and 50 were labeled as Mid *PSE* and Mid *PV*, and students below percentile 25 were labeled as Low *PSE* and Low *PV*.

Multiple regression models were built with each *PSCORE* as response variable, and both *PV* and *PSE* as predictors. The regression R-squared values ranged between 0.0 and 0.1 , and regression coefficients were found non-significant. In spite of this, we observed that students with High *PV* and High *SE* had a notable academic performance advantage in time block 3, i.e., the module with the highest failure rate in the entire course, over students below that mark (see Figure 3). In time block 3, students in module 3 with Low *PSE* (i.e., 17) had a pass

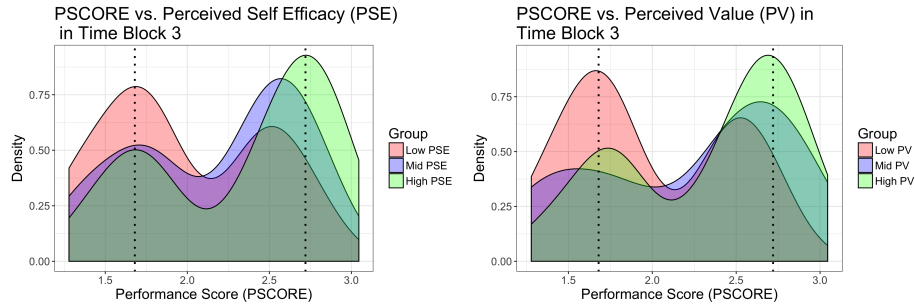


Fig. 3. Density plots for *PSE* (left) and *PV* (right) groups, vs *PSCORE3*. Dashed lines show the cut *PSCOREs* of students passing modules 2 and 3.

rate of 0.118 . Constrastingly, students with Mid *PSE* (i.e., 21) had a pass rate of 0.238 and students with High *PSE* (i.e., 40) had a pass rate of 0.525 . The chance of failing was 48% greater in students below percentile 50 than those above it. A Kruskal-Wallis test for differences in mean *PSCORE3* among *PSE* groups in module 3 yielded a statistically significant result ($\chi^2(2) = 11.298, p < 0.01$). In time block 2, differences among groups were found to be not as substantial, as students with High *PSE*, Mid *PSE* and Low *PSE* had a pass rates of 0.77 , $.75$ and $.721$, respectively. With *PV*, results in modules 2 and 3 were similar to those considering *PSE*, that is, students with High *PV* in time block 3 and module 3, had a pass rate of 0.439 , compared to students with Low *PV*, who had a pass rate of 0.105 .

6 Conclusions and Future Work

In this research we explored the feasibility of predicting academic performance in an introductory programming course, based on indicators derived from implicit theories of intelligence, error orientation, and student attitudes towards computer programming.

An early measurement of constructs linked to general self-theories of intelligence did not correlate with academic results, nor it offered meaningful information for prediction. This is a similar result to that of recent studies conducted with cohorts of Finnish and Turkish students [15, 16]. Around fifty six percent of students in our sample were found to be markedly incremental theorists, while only six percent of students are entity theorists. In our study there is no evidence that the latter were at a disadvantage in the course against the former.

According to the error strain measurement conducted, female students are likely to have stronger emotional reactions than men (e.g., frustration, displeasure) when making mistakes when learning. Although we did not find a clear relationship between this variable and academic performance in programming, it can be argued that measures of error strain in both extremes may be linked to maladaptive behaviors in presence of programming errors. A low error strain may imply a careless behavior, whereas a high error strain may lead to anxiety and frustration.

The measures of perceived self-efficacy and perceived value at the beginning of module 2 allow us to anticipate which students may have greater difficulties (or aptitude) in module 3. This prompts us to devise interventions to boost students' value perceptions and self-efficacy beliefs early in the course, and to help them sustain and strengthen positive beliefs as the course complexity progresses.

In our future work, we will investigate in a larger, cross-institutional sample of engineering freshmen, and with repeated measures, the predictive potential of variables related to self-regulation and metacognition in introductory programming courses, while also including the perceived self-efficacy and perceived value constructs presented in the current study.

References

1. Beanland, D., & Hadgraft, R. (2013) Engineering Education: Transformation and Innovation. UNESCO Report.
2. Froyd, J. E., Wankat, P. C., & Smith, K. A. (2012). Five major shifts in 100 years of engineering education. *Proceedings of the IEEE*, 100, 1344–1360.
3. Watson, C. & Li, F. W. (2014). Failure rates in introductory programming revisited. In *Proceedings of the 2014 conference on Innovation & Technology in Computer Science Education*, 39–44. ACM.
4. Costa, A., & Faria, L. (2018). Implicit theories of intelligence and academic achievement: a meta-analytic review. *Frontiers in Psychology*, **9**(829).
5. Rybowskiak, V., Garst, H., Frese, M., & Batinic, B. (1999). Error orientation questionnaire (EOQ): Reliability, validity, and different language equivalence. *Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior*, **20**(4), 527–547.

6. Steuer, G., Rosentritt-Brunn, G., & Dresel, M. (2013). Dealing with errors in mathematics classrooms: Structure and relevance of perceived error climate. *Contemporary Educational Psychology*, **38**(3), 196–210.
7. Dorn, B. & Elliott Tew, A. (2015). Empirical validation and application of the computing attitudes survey. *Computer Science Education*, **25**(1), 1–36.
8. 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., & Tutty, J. (2005b) Programmed to succeed?: A multi-national, multiinstitutional study of introductory programming courses. Computing Laboratory Technical Report 1-05, University of Kent, Canterbury, UK.
9. Robins, A. (2010). Learning edge momentum: A new account of outcomes in CS1. *Computer Science Education*, **20**(1), 37–71.
10. Capstick, C. K., Gordon, J. D., & Salvadori, A. (1975). Predicting performance by university students in introductory computing courses. *ACM SIGCSE Bulletin*, **7**(3), 21–29.
11. Cronan, T. P., Embry, P. R., & White, S. D. (1989). Identifying Factors that Influence Performance of Non-Computing Majors in the Business Computer Information Systems Course. *Journal of Research on Computing in Education*, **21**(4), 431–446.
12. Rountree, N., Rountree, J., Robins, A., & Hannah, R. (2004). Interacting Factors that Predict Success and Failure in a CS1 Course, **36**(4).
13. Watson, C., Li, F. W., & Godwin, J. L. (2013). Predicting performance in an introductory programming course by logging and analyzing student programming behavior. In: *IEEE 13th International Conference on Advanced Learning Technologies (ICALT)*, pp. 319–323, IEEE, Beijing, China (2013).
14. Dweck, C. (1999). *Self-theories. Their Role in Motivation, Personality, and Development*. Philadelphia: Psychology Press.
15. Kaijanaho, A.-J., & Tirronen, V. (2018). Fixed versus Growth Mindset Does not Seem to Matter Much. *Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER 18*, 11–20.
16. Tek, F. B., Benli, K. S., & Deveci, E. (2018). Implicit Theories and Self-efficacy in an Introductory Programming Course. *IEEE Transactions on Education*, **61**(3), 218–225.
17. MIT Scratch (2018). <https://scratch.mit.edu/>. Last accessed 7 Dec 2018.
18. NumPy (2018). <http://www.numpy.org>. Last accessed 7 Dec 2018.
19. Zumbo, B. D., Gadermann, A. M., & Zeisser, C. (2007). Ordinal versions of coefficients alpha and theta for Likert rating scales. *Journal of Modern Applied Statistical Methods*, **6**, 21-29.
20. Adams, W. K., Perkins, K. K., Podolefsky, N. S., Dubson, M., Finkelstein, N. D., & Wieman, C. E. (2006). New instrument for measuring student beliefs about physics and learning physics: The Colorado Learning Attitudes about Science Survey. *Physical review special topics-physics education research*, **2**(1), 1–14.