# The influence of motivation and comfort-level on learning to program

Susan Bergin
National University of Ireland, Maynooth

Ronan Gabriel Reilly
National University of Ireland, Maynooth

# The influence of motivation and comfort-level on learning to program

Susan Bergin and Ronan Reilly

Department of Computer Science, NUI Maynooth
$\{sbergin, ronan\}@cs.nuim.ie$
WWW home page: http://www.cs.nuim.ie

**Abstract.** This paper documents a study, carried out in the academic year 2004-2005, on the role of motivation and comfort-level in a first year object-oriented programming module. The study found that intrinsic motivation had a strong correlation with programming performance as did self-efficacy for learning and performance, r=$0.512$, $p < 0.01$ and r=$0.567$, $p < 0.01$ respectively. Aspects of comfort level were found to have significant correlations with performance with an instrument on programming-esteem rendering the most interesting results. A regression model based upon these factors was able to account for $60\%$ of the variance in programming performance results.

## 1 Introduction

It is well known in the Computer Science Education (CSE) community that students have difficulty with programming courses and this can result in high drop-out and failure rates. This is a cause of great concern for educators and has led to a body of research in the area. Although many studies have interesting results it can be hard to apply the results to other educational settings with different parameters. Furthermore, while a considerable amount of research has been carried out on factors that affect programming performance, our interest is solely in factors that affect performance on an introductory third-level object-oriented programming module, where such factors can be determined early in the academic year. The research documented in this paper is part of a longitudinal study on factors that influence performance on introductory programming modules. In our earlier work [2] we examined the effect of fifteen factors on programming. In this paper we report on two further cognitive factors: motivation and comfort-level.

Student motivation can be described as a student's willingness, need, desire and compulsion to participate and be successful in the learning process [4]. Although students may be equally motivated to perform a task, the sources of their motivation may differ. A student who is intrinsically motivated undertakes an activity for its own sake, for the enjoyment it provides, the learning it permits, or the feelings of accomplishment it evokes. An extrinsically motivated student is motivated by some reward or by avoiding some punishment external to the activity itself, such as grades [13]. A number of studies e.g. [14], have found that students who are more intrinsically motivated than extrinsically motivated perform better, and that using extrinsic motivators to engage students in learning can both lower achievement and negatively affect student motivation.

Our definition of comfort-level incorporates a student's ease when asking and answering programming questions, their self-perception of how they are doing, their self-esteem and self-efficacy for programming. Self-esteem is a positive or negative orientation towards oneself. When applied to programming it can be understood as a positive or negative orientation to programming. Self-efficacy can be described as a person's judgement of their capabilities to organise and execute courses of actions required to attain particular types of performance. The most important source of self-efficacy is the individual's evaluation of the outcomes of his or her direct attempts to perform an activity [1].

## 2 Background

A number of studies have examined the importance of motivation in programming. Some researchers have focused on the reasons why students choose to study programming and how these reasons change over time, including [6], [11], [16]. In one such study it was found that $22\%$ of a first year introductory programming class were studying programming because they were interested in it, $40\%$ because they wanted to obtain or change jobs and $35\%$ because it was a requirement. The remaining $5\%$ did not respond [16]. In another similar study it was found that aspiration for some future gain (possibly financial) and desire to learn accounted for approximately $76\%$ of the motivations for choosing a computer science degree. $50\%$ of all students responded that their reason for taking an introductory programming module because it was compulsory with only $20\%$ indicating that they were interested in learning programming. Other researchers have examined how motivation on programming modules changed over time [6]. It was found that the major motivation when students' start programming courses is to become a professional programmer but this is not the case when the course has advanced. Although programming is continually regarded as useful it is seen more as a secondary skill later on. Unfortunately, none of the authors correlated the reasons for studying programming with performance on the course and thus their results, while interesting, are limited for our purposes.

Some researchers have considered the relationship between student motivation and impressions of computing subjects. It has been found that students who have positive impressions of a subject tended to be both intrinsically and extrinsically motivated. Furthermore, students who feel they have a strong motivation for studying a subject have a more positive perception of the subject and about the amount of practical work involved, the clarity of the subject matter and their final grades [18].

Other researchers have focused, for example, on aspects of programming and technologies that can be used as motivators. It has been suggested that programming exercises should be improved to be more motivating [24] and that particular programming languages should be used for a first programming language based on related motivational factors [8]. The use of web and game programming examples in place of classical programming examples have been found to be more motivating on first programming courses [7]. Competitive programming, where students develop and improve their code throughout an assignment by competing in a tournament against instructor-defined code and the code of other students has been introduced into an introductory programming module in an attempt to increase motivation [12]. Some academics have gone even further and have introduced a new evolutionary introductory programming course specifically designed to improve students experiences and to improve retention [15].

In recent years a number of studies have examined the role of comfort-level in programming. It has been suggested that while female and male students report similar levels of comfort in using computers, women assessed their peers on the course as being more comfortable with computers than they were while men assessed themselves, on average, as slightly more comfortable than their peers [9]. In another study the relationship between attitude to programming and performance on an introductory programming course was examined. Programming confidence levels were found to be important in students participating in pair programming activities, most notably in that students who were very confident did not enjoy the experience of pair programming as much as other students, and that students produced their best work when placed in pairs with students of similar confidence levels [25]. Researchers have also examined the relationship between students' expectations of and experiences on an introductory computing module. A positive relationship has recently been identified between students' mental models of programming and self-efficacy for programming and performance [26]. In addition, it has been found that the grade a student expected to achieve in an introductory module was the most important indicator of performance [23]. Finally, a recent longitudinal study on factors that influence programming success found that the most important predictor of students' performance on an introductory computer science course was comfort level, determined by the degree of anxiety a student felt about the course [5].

In summary, research on student motivation in programming has considered a diverse range of topics including why students' choose to study programming, the effect of different types of motivation (intrinsic and extrinsic) on a students perception of computing and efforts to make courses more motivating. While these studies are interesting and insightful they fail to answer the research questions we are interested in. Additionally, research on the role of comfort-level on programming performance has proved fruitful and we endeavor to built upon the existing studies to evaluate the following research questions: (1) what are the effects of intrinsic and extrinsic motivation on student performance on a first year programming module, (2) what is the effect of comfort level on student performance on a first year programming module and (3) are motivation and comfort-level either singularly or additively useful factors for predicting performance on a first year programming module.

## 3 Research Methodology

The introductory programming module at our university is composed of a one and a half hour Problem-Based Learning (PBL) workshop [10], a one and a half hour laboratory session and three one-hour lectures every week over two semesters. In general, students in Ireland do not study programming in secondary school and the majority of students taking this module have recently completed second level education. Performance on this module is based on continuous assessment ($30\%$ of the overall mark) and a final examination ($70\%$ of the overall mark).

The study was carried out in the academic year 2004-2005. Students enrolled in the first year 'Introduction to Programming' module in our department voluntarily participated in this study. Approximately, 110 students are currently registered for this module. The measure of performance presented in this paper is the results (student scores) on an early lab examination.

**Table 1.** Research questions and associated hypotheses

| **Research questions** | **Hypotheses** |
|---|---|
| (1) What are the effects of intrinsic and extrinsic motivation on student performance on a first year programming module? | (1.a) Students with higher levels of intrinsic motivation will perform better in programming than students with lower levels of intrinsic motivation.<br>(1.b) There will be a difference in the programming performance of students based on their degree of extrinsic motivation.*<br>(1.c) Students with a higher degree of intrinsic motivation will perform better in programming than students with a higher degree of extrinsic motivation. |
| (2) What is the effect of comfort level on student performance on a first year programming module? | (2.a) Students with higher scores on the nine in-house questions will perform better in programming than students with lower scores on these questions.<br>(2.b) Students with higher scores on the self-esteem scale will perform better in programming than students with lower scores.<br>(2.c) Students with higher scores on the self-efficacy for programming questionnaire will perform better in programming than students with lower scores. |
| (3) Are motivation and comfort-level either singularly or additively useful factors for predicting performance on a first year programming module? | To determine the predictive ability of motivation and comfort level a number of regression analysis were performed on each of the factors and their component parts. |

* The authors did not attempt to hypotheses on the direction of the relationship between extrinsic motivation and programming performance.

To address each of the research questions in this study a number of hypotheses were developed. The research questions and corresponding hypotheses are outlined in Table 1.

Two instruments were used to collect data: a motivation questionnaire and a background questionnaire. The Motivated Strategies for Learning Questionnaire (MSLQ) was developed at the National Center for Research to Improve Postsecondary Teaching and Learning at the University of Michigan. It was designed to assess the motivational orientations and use of learning strategies of college students [19]. It is based on a cognitive view of motivation and is composed of two sections: a motivation section and a learning strategies section. This paper only reports upon the motivation section. The motivation section consists of 31 questions on a 7-point scale (1=not at all true of me, 7=very true of me) and consists of six sub-scales that measure:

- Intrinsic goal orientation
- Extrinsic goal orientation
- Task value (a student's perceptions of the course material in terms of interest, importance and utility)
- Control of learning beliefs (a student's beliefs that outcomes are contingent on ones own effort, in contrast to external factors such as the teacher)
- Self-efficacy for learning and performance (incorporates two aspects of expectancy: expectancy for success and self-efficacy)
- Test anxiety

Cronbach's alphas for each of the sub-scales are given as 0.74, 0.62, 0.90, 0.68, 0.93 and 0.80 respectively [19]. In this study the calculated alpha coefficients were: 0.73, 0.33, 0.84, 0.71, 0.96, and 0.80.

The background questionnaire was composed of two sections. The first section gathered information on previous academic and non-academic information and the second section gathered information on programming comfort-level.

This paper only considers the responses to the comfort-level section. The comfort-level section is composed of three sub-sections: (1) a number of questions examining a student's perception of their programming experience, (2) an adapted Rosenberg Self-Esteem questionnaire [22], and (3) a shortened version of the Computer Programming Self-Efficacy Scale [20]. Each of the sub-parts are described further below.

Based on the earlier work of [5] a set of nine questions on student experience of programming were included. The questions examined a student's perception of their level of understanding compared to the rest of the class (one question), their ease at asking and answering programming questions (five questions), their general understanding of programming concepts and their ability to design and complete assignments (three questions). Tests of reliability using Cronbach's alpha were 0.77 in this study for the nine student experience questions.

The Rosenberg Self-Esteem (RSE) questionnaire was adapted to apply to programming-esteem. The RSE scale is perhaps the most widely used self-esteem measure in social science research. The scale consists of ten questions and has been shown to generally have a high reliability: test-retest correlations are typically in the range of 0.82 to 0.88, and Cronbach's alpha for various samples are in the range of 0.77 to 0.88 [21]. Each of the questions were re-worded to relate to programming esteem and not to self-esteem directly, for example the first question was changed from ‘On the whole, I am satisfied with myself’ to ‘On the whole, I am satisfied with my Java programming progress’. The Cronbach alpha coefficient for this study was found to be 0.61, however the omission of item 7 ‘I feel that I’m a person of worth, at least on an equal plane with the other Java programmers in my class’ from the scale resulted in an alpha of 0.93. Given the poor correlation between this item and all other items on the scale, responses for this item were not included in the analysis documented in this paper.

The Computer Programming Self-Efficacy Scale consists of thirty-three items that ask students to judge their capabilities in a wide range of programming tasks and situations. Responses are answered on a 7-point scale (1=not at all confident, 7=absolutely confident). We used a shortened version of this scale consisting of only eight questions about simple programming tasks. Cronbach’s alpha in this study was 0.89.

In total 57 students (38 male, 19 female) completed the background survey and 33 students (20 male, 13 female) completed the motivation survey. Both surveys were completed in the first semester, after four and ten weeks of programming labs respectively.

## 4 Results

An *a priori* analysis was carried out to verify no significant differences existed between the mean module scores of the class and the sample on the lab exam. A t-test $(t(173) = 0.255, p = 1.346)$, revealed no significant differences between the class and the sample scores. Although test assumptions of equality of variance were confirmed using the Levine test, Kolmogorov-Smirnov tests revealed that assumptions of normality were violated on the class and sample scores. The distributions were found to be negatively skewed and negatively kurtic. Although the deviation was not considerable, two independent transformations were performed on the lab scores (square root and arcsine transformations). The statistical analysis reported in the following section was carried out on the raw lab scores and also on the transformations. In all instances, the correlations and subsequent regression models derived using the raw scores were found to be highly similar to those of the transformed scores. Given this, only the results of the statistical analysis carried out on the raw lab scores are reported on in this paper however the results found using the transformations are detailed in [3]. In the remainder of this section the findings on each of the hypotheses studied is presented, and the combination of factors that best predicts performance is described.

### 4.1 Motivation

To analyze the relationship between motivation and programming performance students were grouped based on their responses to questions on the intrinsic and extrinsic goal orientation scales of the MSLQ. Students were grouped using three separate methods, reflecting each of the three separate hypotheses, 1a, 1b and 1c according to their degree of: intrinsic goal orientation (high, medium or low level of intrinsic goal orientation); extrinsic goal orientation (high, medium or low level of extrinsic goal orientation); and both intrinsic and extrinsic goal orientation (high level of both intrinsic and extrinsic goal orientation, medium level of intrinsic and extrinsic goal orientation, low level of intrinsic and extrinsic goal orientation, higher level of intrinsic than extrinsic goal orientation and higher level of extrinsic than intrinsic goal orientation respectively.

**Hypothesis 1a:** *Students with higher levels of intrinsic motivation will perform better in programming than students with lower levels of intrinsic motivation.*
**Results:** As only one student was defined to have low intrinsic motivation no statistics were generated on this category. As there were only two other groups an independent t-test was used to determine if any statistical difference existed between the mean lab exam scores of students with high levels of intrinsic motivation and the mean lab exam scores of students with medium levels of intrinsic motivation. The mean score of students with a high level of intrinsic goal orientation was found to be significantly higher than that of students with a medium level of intrinsic goal orientation $(t(39) = 3.368, p = 0.002)$.

**Hypothesis 1b:** *There will be a difference in the programming performance of students based on their degree of extrinsic motivation.*
**Results:** As only one student was defined to have low extrinsic motivation an independent t-test was used to determine if any statistically significant difference existed between the mean scores of students with a high level of extrinsic motivation and the mean scores of students with a medium level of extrinsic motivation. The t-test found no statistically difference between the two groups.

**Hypothesis 1c:** *Students with a higher degree of intrinsic motivation will perform better in programming than students with a higher degree of extrinsic motivation.*
**Results:** A one-way ANOVA revealed a significant difference between the mean scores of students based on their degree of combined intrinsic and extrinsic motivation $(F(4, 27) = 3.703, p = 0.016)$. As only one student was found to have a low level of both intrinsic and extrinsic motivation this group had to be omitted from post hoc analysis. Tukey HSD, Scheffe and Bonferroni tests failed to reveal the cause of the differences. Given that we had reason to believe that an important difference exists based on motivational types we carried out further analysis using the less rigorous Least Significant Difference test (LSD). While we acknowledge the increased risk of a Type I error, it is worthwhile noting that the LSD test revealed that the mean score of students with a high level of both intrinsic and extrinsic goal orientation was significantly higher than that of students with a higher level of extrinsic than intrinsic goal orientation and that students with higher levels of intrinsic than extrinsic goal orientation were found to have a higher mean score than students with a higher levels of extrinsic than intrinsic goal orientation.

To supplement our findings Pearson correlations for each of the individual MSLQ questions, the total score on MSLQ and the total score for each of the six sub-categories were generated and analyzed. The total score for MSLQ and for the sub-categories intrinsic goal orientation, task value, control of learning beliefs and in particular self-efficacy for learning and performance all had a statistically significant relationship with performance on the course, as depicted in Table 2. None of the questions on test-anxiety, or extrinsic goal orientation had significant correlations.

**Table 2.** Significant Pearson correlations for MSLQ total score and sub-sections and performance

| Measure | N | Lab Test Results |
|---|---|---|
| Total for MSLQ | 33 | .494(**) |
| Total Intrinsic Goal Orientation | 33 | .512(**) |
| Total Task Value | 33 | .438(**) |
| Total Control of Learning Beliefs | 33 | .300(*) |
| Total Self-Efficacy for Learning and Performance | 33 | .567(**) |

** Correlation is significant at the 0.01 level (1-tailed).
* Correlation is significant at the 0.05 level (1-tailed).

### 4.2 Comfort Level

**Hypothesis 2a:** *Students with higher scores on the nine in-house questions will perform better in programming than students with lower scores on these questions.*
**Results:** Students were grouped according to their responses to each of nine in-house questions. Each question had five possible answers from 1 to 5 inclusive, where 1 indicated being very uncomfortable and 5 indicated being very comfortable. For each student the number of responses in each category (1,2,3,4,5) were counted and mapped to an ‘experience on the module’ scale of high, medium or low. A one-way ANOVA failed to reveal any statistically significant differences between the mean score of any of the student groups.

**Table 3.** Significant Pearson correlations for comfort level and performance

| Measure | N | Lab Test Results |
|---|---|---|
| **Total shortened Self Efficacy scale** | 54 | .321(**) |
| **Total Adapted-Self Esteem scale** | 54 | .361(**) |
| **In-house questions** | | |
| Level of understanding | 55 | .368(**) |
| Course concepts | 55 | .273(*) |
| Designing a program | 55 | .296(*) |

** Correlation is significant at the 0.01 level (1-tailed).
* Correlation is significant at the 0.05 level (1-tailed).

**Hypothesis 2b:** *Students with higher scores on the self-esteem scale will perform better in programming than students with lower scores.*

**Results:** Students were sorted into groups reflecting their level of programming esteem (high, medium and low).A one way ANOVA revealed a difference between the mean scores of students based on self-esteem $(F(2,50) = 3.223, p = 0.048)$ and subsequent analysis using Tukey HSD found that this difference existed between students with high levels of self-esteem and students with low levels of self-esteem.

**Hypothesis 2c:** *Students with higher scores on the self-efficacy for programming questionnaire will perform better in programming than students with lower scores.*
**Results:** Again, the students were divided into groups reflecting their level of self-efficacy (high, medium and low). A one-way Anova failed to reveal any statistical differences between the mean scores of students based on their self-efficacy, however, the results did show a tendency towards significant, $(F(2,51) = 3.075, p = 0.055)$, the results cannot be dismissed entirely.

As with motivation we carried out secondary analysis using Pearson correlations to further examine the relationship between comfort level and programming performance. The total on the adapted RSE and on the computer programming self-efficacy scale was found to have a statistically significant relationship with performance while the total for the nine in-house questions had not. However, of the nine in-house questions only three rendered significant correlations with performance, specifically: (1) a students perception of their understanding of the programming module compared to the rest of the class, (2) their understanding of programming concepts and (3) their ability to design programs without help. Pearson correlations are given in Table 3.

### 4.3 Regression Analysis

**Hypothesis 3:** *To determine the predictive ability of these factors a number of regression analysis were performed on each of the factors and their component parts.*
**Results:** To investigate whether the various factors studied were predictive of performance on the module a number of regression models were developed. Each model was motivated by the literature review, the authors' experience working with first year students and the strength of the correlation coefficients generated in this study.

In the first model consideration was given to the total score for each of the MSLQ sub-categories and the comfort level sub-categories. Using a stepwise regression method a significant model emerged with $F(1,31) = 14.710, p < 0.001$ with an adjusted R square =30$\%$ for the lab test results. The only significant variable in the model was the total self-efficacy for learning and performance score from the MSLQ. Given the importance of intrinsic motivation, a regression model was developed based upon student's intrinsic goal orientation levels. The model resulted in an adjusted R square = 33$\%$ with $F(1,30) = 16.364, p < 0.001$. Expanding this model to incorporate the total scores of the sub-categories in comfort level, resulted in a regression model, with an adjusted R square = $40\%$, $F(2,27) = 10.845, p < 0.001$, with two significant variables, the intrinsic goal orientation grouping and students grouped based on their programming self-esteem. Considering the responses to each of the self-esteem questions and intrinsic goal orientation accounted for a higher variance in the scores, with an adjusted R square = $53\%$, $F(2,27) = 17.218, p < 0.001$ based upon question 4: 'I am able to do

Java programming tasks as well as most other students in the class' and the intrinsic goal orientation grouping. Finally, a regression model that considered the responses to each of the individual questions (as opposed to total scores in each category) accounted for $60\%$ of the variance in scores, $F(4, 27) = 22.987, p < 0.001$, for three motivational questions and for ease of asking questions in lectures.

## 5 Discussion

Our results partially support the findings of previous studies, for example [14], that students who are more intrinsically motivated than extrinsically motivated perform better. Furthermore, it appears that the level of intrinsic motivation has a positive impact on performance, with higher levels of intrinsic motivation resulting in higher programming results. The role of extrinsic motivation when considered on its own, seems to have little effect on programming performance and no statistically significant correlation could be established. It would appear that while intrinsic goal orientation affects programming performance extrinsic motivation does not. Given this, one must consider whether the use of grades, rewards or student comparisons are at all useful measures for motivating programming students and indeed if research efforts should focus on re-evaluating techniques to better foster intrinsic motivation.

The other scales on the MSLQ provided some useful results. The value students associate with a task in terms of how important, how useful and how interesting it is had a significant positive correlation with programming performance. Students beliefs that their efforts to learn will result in positive outcomes (control of learning beliefs) was also found to have a significant relationship with performance.

Furthermore, the strong results generated on the self-efficacy of learning and performance section on the MSLQ supports our reasoning for studying comfort level. The three instruments used to collect data on comfort level have proved useful and the development of a single instrument based on a refinement of these could result in more significant findings in the future. As with most exploratory research studies, the main benefit is that the results provide pointers for more focused research work and in this case certainly encourage further studies.

## 6 Conclusions

This study examined the relationship and predictive ability of motivation and comfort level on programming performance. Although our findings are interesting, the results are limited as the number of students who participated in the study was relatively small (only $30\%$ of the class completed the MSLQ) and the findings are based on a single trial. However, we feel that the positive results from this initial investigation warrant further research. To this end we propose to repeat this study on a larger scale to establish our findings.

## References

1. Bandura A.: Social foundations of thought and action : a social cognitive theory. Englewood Cliffs, N.J. : Prentice-Hall, 1986.

2. Bergin, S., Reilly R.: Programming: Factors that influence success. SIGCSE 2005. Proceedings of the thirty-fifth SIGCSE technical symposium on Computer Science Education. St. Louis, Illinois, US. February 2005.
3. Bergin, S., Reilly R.: Do Motivation and comfort-level influence programming success? NUIM Technical Report Series, March 2005.
4. Bomia, L., Beluzo, L., Demeester, D., Elander, K., Johnson, M., & Sheldon, B.: The impact of teaching strategies on intrinsic motivation. Champaign, IL: ERIC Clearinghouse on Elementary and Early Childhood Education. (ERIC Document Reproduction Service No. ED 418 925), 1997.
5. Cantwell-Wilson B., Shrock S.: Contributing to success in an introductory computer science course: a study of twelve factors. Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education. pg184-188. 2001.
6. Curzon P., Rix J.: Why do Students take Programming Modules? Proceedings of the 6th annual conference on the teaching and computing and the 3rd annual conference on integrating technology into CSE: Changing the delivery of Computer Science Education. ITICSE ’98. Dublin, Ireland, pg 5963. 1998.
7. Feldgen M., Clua O.: New Motivations are Required For Freshman Introductory Programming. 33rd ASSE/IEEE Frontiers in Education Conference. Boulder, Co. Vol. 1, T3C-T24. November 2003.
8. Hadjerrouit S.: Java as First Programming Language: A Critical Evaluation. SIGCSE Bulletin. Vol. 30. No 2. pg 43-47. June 1998.
9. Irani L.: Understanding Gender and Confidence in CS Course Culture. SIGSCE ’04. Norfolk. Virginia USA. Pg 196-199. March 2004.
10. O’Kelly. J, Bergin. S, Ghent. J, Gaughran. P, Dunne. S, Mooney. A.: An Overview of the Integration of Problem Based Learning into an Existing Computer Science Programming Module. PBL International Conference, Mexico, June 2004.
11. Jenkins T.: The Motivation of Students in Programming. ITICSE’01. Canterbury, UK, pg 53-56. June 2001.
12. Lawrence R.: Teaching Data Structures Using Competitive Games. IEEE Transactions on Education Vol. 47 No 4. November 2004.
13. Lepper, M. R.: Motivational Considerations in the Study of Instruction. Cognition and Instruction. Vol. 5. Issue 4 pg. 289-309. 1988.
14. Lumsden, L.S.: Student motivation to learn (ERIC Digest No. 92). Eugene, OR: ERIC Clearinghouse on Educational Management. (ERIC Document Reproduction Service No. ED 370 200). 1994.
15. Mahmoud Q.H., Dobosiewicz W., Swayne D.: Redesigning Introductory Computer Programming with HTML, JavaScript and Java. Proceedings of the 35th SIGCSE technical symposium on computer science education. SIGCSE’04. Norfolk Virginia, USA. pg 120-124. March 2004.
16. Mamone S.: Empirical Study of Motivation in an Entry Level Programming Course. ACM SIGPLAN Notices. Vol. 27, No 3, pg 54-60. March 1992.
17. McKeachie W.J.: McKeachie’s Teaching Tips: Strategies, Research, and Theory for College and University Teachers. 11th edition. Houghton-Mifflin Higher Education. 2002.
18. Mitchell M., Shearad J., Makham S.: Student Motivation and Positive Impressions of Computing Subjects. ACM International Conference proceedings series. Proceedings of the Australasian conference on computing education. Pg 189 -104ACM 2000.
19. Pintrich P., Smith D., Garcia T., McKeachie W.: A Manual for the Use of the Motivated Strategies for Learning Questionnaire. Technical Report 91-B-004. The Regents of the University of Michigan. 1991.

20. Ramalingam V., Wiedenbeck S.: Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. Journal of Educational Computing Research, Vol. 19 Issue 4. pg 365-379. 1998.
21. Rosenberg, Morris. Conceiving the Self. Krieger: Malabar, FL. 1986.
22. Rosenberg M.: Society and the adolescent self image. Princeton, NJ: Princeton University Press. 1965.
23. Rountree N., Rountree J., Robins A.: Predictors of success and failure in a CS1 course .SIGCSE Bulletin. Volume 34. No 4. pg 121-124. 2002.
24. Tharp A. L.: Getting More Oomph from Programming Exercises. Proceedings of the 12th SIGCSE Technical Symposium on Computer Science Education. St. Louis, Missouri. US. ACM SIGCSE Bulletin Vol. 13 Issue 1 pg 91-91. February 1981.
25. Thomas L., Ratcliffe M., Roberstson A.: Code Warriors and Code-a-Phobes: A Study in Attitude and Pair Programming. Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education. SIGCSE'03. Ren Nevada. ACM SIGCSE Bulletin. Vol. 35 Issue 1. pg 363-367. February 2003.
26. Wiedenbeck S., LaBelle D., Kain V.: Factors affecting course outcomes in introductory programming. Proceedings of the 16th Workshop on Psychology of Programming. PPIG'04. pg. 97-109. 2004.