Visual Basic Programming Impact on Cognitive Development of College Students

Garry White
Computer Information Systems Department
Texas State University
600 University Drive,
San Marcos, TX 78666
gw06@txstate.edu

ABSTRACT

This research investigated the impact learning a visual programming language, Visual Basic, has on cognitive development, as measured by the Proposition Logic Test (PLT). It is part of a series of cognitive and programming investigations. Prior research has shown that object-oriented and procedural programming involved a high level of cognitive development. Prior research also indicated procedural programming has no impact on cognitive development. This study found cognitive development remained the same after a semester course in Visual Basic. The author concludes that a single college programming course, either procedural or visual, does not impact college age students' cognitive development.

Keywords: Visual programming, Visual Basic, Cognitive development, Cognitive skills.

1. INTRODUCTION

1.1 Overview

Should all students learn programming? Soloway (1993) says programming enables the learning of various subject areas and develops new ways of looking at the world. In 1984, pre-high school programming was taught because higher cognitive skills would be impacted (Pea & Kurland, 1984). Literature suggests that computer programming does enhance cognitive skills such as reasoning, problem solving, and logic thinking for high school and elementary school students (Bernardo & Morris, 1994; Palumbo & Reed, 1991; Seidman, 1989-90). And computer programming has been shown to support algebraic concepts (Hart, 1983).

At the same time, literature has shown computer programming does not improve mathematical problem-solving skills (Flores, 1985; McCoy, 1987; Platt, 1990). Nor does it impact elementary school children's cognition (Clements & Gullo, 1984). A belief that curriculum can impact cognitive development may be misleading.

The weaknesses in these studies are a lack of focus with the cognitive development level of the students (Bernardo & Morris, 1994) and how cognitive development is different from cognitive skills. This research defines cognitive development as levels of abstract/concrete thinking in terms of Piaget's cognitive development theory. It indicates what "can" be learned. Skills are defined as methods/procedures learned. Skills indicate what "is" learned. Such learned skills can show the level of cognitive development a person is at. For example, FORTRAN programming can increase problem solving skills, yet subjects are already at the highest level of cognitive development (Choi & Repman, 1993).

Research has shown computer programming relates to cognitive development and can improve cognitive skills (Bernardo & Morris, 1994; Cafolla, 1987; Evans & Suimkin, 1989; Fletcher, 1984; Gibbons, 1995; Ignatuk, 1986; Little, 1984; Palumbo & Reed, 1991; Seidman, 1989-90). The question is whether computer programming can enhance cognitive development; move from concrete thinking to abstract thinking or improve abstract thinking. Research has shown procedural programming courses do not improve cognitive development (Ignatuk, 1986; Mains, 1997; Owens & Seiler, 1996; Priebe, 1997). However, procedural programming deals with a high level of cognitive development. A theory by White & Sivitanides (2002) suggests visual programming deals with a lower level of cognitive development due to visual objects. Visual objects provide opportunities for concrete thinking.

There has been no research dealing with the impact on the cognitive development of new programming languages, such as Visual Basic v6, that contain visual object concepts. This is the purpose of this paper; to investigate the impact learning a visual programming language, Visual Basic v6, has on cognitive development as measured by the Proposition Logic Test (PLT). A prior study has shown a relationship between the PLT and Visual Basic v6 (White & Ploeger, 2004).

1.2 Importance of Study:

Prior cognitive research in programming has been with procedural and object-oriented languages, such as Logo, Basic, Pascal, FORTRAN, C++, and Java. Since Visual programming uses visual objects on a screen, lower cognitive development might be advanced where procedural languages have failed. Understanding the impact of computer programming languages on cognitive development will provide better cause/effect research, curriculum adjustment, and advising of students. The results from this study contribute to the knowledge of programming language impact on cognitive development.

2. LITERATURE REVIEW

2.1 Piaget's Cognitive Development Theory

Piaget's cognitive development occurs in three stages (Epstein, 1990; Fischer & Silvern, 1985; Piaget, 1972). The 3 stages are (Beihler & Snowman, 1986):

1. Preoperational – work with visual symbols such as words & unable to reverse actions.
2. Concrete – can reverse actions; understand physical objects and past experience, can only generalize concrete experiences.
3. Formal – form hypotheses, solve problems, deal with and manipulate abstract ideas, and develop logic reasoning.

Piaget's theory indicates that around middle school, ages 11-12, formal thinking abilities develop (Chiapetta, 1976; Lovell, 1961). At this age, students move from concrete thinking to formal thinking. These students begin to think logically and use abstractions.

However, research has shown that Formal Operational thinking is reached at different ages or not at all (Bastian et al., 1973; Epstein, 1980; Griffiths, 1973; Renner et al., 1978; Schwebel, 1975). In adulthood, cognitive development becomes fixed (Kuhn et al., 1977). Maturation may have occurred.

Studies have shown a majority of adults and college students fail at many Formal Operational tasks (Bastian et al., 1973; Griffiths, 1973; Petrushka, 1984; Schwebel, 1972; Sund, 1976). Why? The reason is related to the maturing neural fibers between the left and right cerebral hemispheres (Kraft, 1976). And Ross (1982) found Epstein's physical descriptions of brain growth spurts and plateaus corresponded to Piaget's development stages. The advancement of people through the development of Piagetian stages is an indication of brain development. Maturation may be reached at different cognitive development levels and at different ages.

2.2 Cognitive Development/Abilities Component of Computer Programming

Research has shown that procedural and object-oriented programming required high levels of cognitive development (Cafolla, 1987; Fletcher, 1984; Gibbons, 1995; Ignatuk, 1986; Little, 1984; Monfort et al., 1990; Ott, 1989; Sein & Bostrom, 1989; Wu, 1993; White, 2002). Those at lower cognitive levels have difficulty in learning programming. Becker (1982) found pre-middle school students had difficulty learning procedural programming. This age group had conceptual and representational difficulty in constructing what a computer is doing (Pea & Kurland, 1984). It is not surprising that concrete thinkers are unable to do programming in light of Piaget's theory of cognitive development.

Research has shown that higher cognitive abilities, such as Piaget's formal operations, are involved with procedural programming (Dalbey & Linn, 1985; Hudak & Anderson, 1990). Such abilities can predict procedural programming achievement (Irons, 1982; Ricardo, 1983). Cafolla (1987), Barker (1985), and Azzedine (1987) found that Piaget's cognitive development predicted procedural programming performance.

Little (1984) and Hudak & Anderson (1990) found that students who were Piaget's formal operation thinkers, scored higher on programming and logical thinking then those who were Piaget's concrete/transitional thinkers (Hudak & Anderson,1990; Little, 1984). Students at the higher cognitive development have a greater abstract learning style that enables them to learn programming.

In computer science, the use of logic has been identified (Galton, 1992; Gibbs & Tucker, 1986; Myers, 1990; Sperschneirder & Antoniou, 1991). Part of Piaget's formal operational development level is the ability to do propositional logic (Brainard, 1978; Enyeart, 1981; Inhelder & Piaget, 1958). Propositional logic involves truth tables. IF/THEN/ELSE programming statements deal with such concepts. Propositional logic has a direct analogy in procedural programming (Folk, 1973). An instrument that measures Piagetian tasks of logic is the Propositional Logic Test (PLT). Research has shown that the PLT can predict procedural programming and object oriented programming success (Stager-Snow, 1985; White, 2002; White & Sivitanides, 2005), as well as Visual programming (White & Ploeger, 2004). The question is whether a visual programming course effects cognitive development as indicated by the PLT.

2.3 Procedural Programming non-impact on cognitive development

Research has shown no impact on formal operations thinking skills from taking a procedural programming course or a computer science logic course (Ignatuk, 1986; Kim, 1995; Mains, 1997; Owens & Seiler, 1996; Piburn, 1989). Such research used high school and college age subjects. Cognitive maturation may have been reached.

However, learning to program in Logo, a pseudo programming language, seems to enhance biconditional reasoning, a precondition to Formal Operational thinking, for fifth graders (Seidman, 1989-90). Another study with Logo programming showed the reasoning skills of Junior High School students improved (Many et al., 1988). There was a difference between the 7th grade and 8th grade students. Normal maturation could not be ruled out. It is at this age that formal operational cognitive skills, such as propositional logic thinking, start to develop. The subjects in other studies, where no improvement occurred, were college students over the age of 18. When Logo was taught to 6-year-olds, no differences were found on measures of cognitive development (Clements & Gullo, 1984). Learning to program does not improve cognitive skills for young children who are still in the concrete level of cognitive development (Shaw, 1984).

A study by Stockwell (2002) did find that a course in "C" programming did improve mathematical skills for college students, suggesting that problem solving was enhanced. It should be noted that "C" programming is based on Algebra concepts and uses functions. The pre & post mathematics tests contained such items. The improvement may have been due to learning Algebra concepts and functions using "C" programming. In other words, "C" programming was another way to teach Algebra concepts and functions. These students may already have been at a high cognitive development level.

2.4 Summary

Since cognitive development is fixed in adulthood (Schwebel, 1972) and not all adults develop to Formal Operations (Bastain, et al., 1973; Griffiths, 1973; Schwebel, 1975), a programming language that requires formal cognitive development thinking skills, may do nothing for adults who are at a lower cognitive development level. One college programming course may be too late.

It is theorized that Visual programming requires a lower cognitive development level due to visual components in the language (White & Sivitanides, 2002). Visual objects accommodate concrete thinking. At the same time, visual programming uses procedural programming type modules. Can a visual programming course impact cognitive development to a higher level for college students? Research indicates the answer is "no" for procedural programming with college age subjects (White & Sivitanides, 2002). Procedural programming requires a higher cognitive development level (Mains, 1997; Owens & Seiler, 1996; Priebe, 1997). Visual programming has never been studied as improving logical thinking skills or formal operational cognition. Such research is warranted.

3. METHODOLOGY

3.1 Null Hypotheses

Hypothesis: A Visual Basic v6 programming course does not effect cognitive development, as measured by the Propositional Logic Test (PLT).

3.2 Instruments

The **Propositional Logic Test (PLT)** was developed at Rutgers University. It measures Boolean logical thinking ability, the ability to interpret truth function operations with a stated rule. The PLT requires 15 minutes to administer. The subject responds to 16 items, which measure Piagetian formal operations. The PLT shows the ability to deal with IF/THEN/ELSE programming statements. The PLT has been used to study logical reasoning as a predictor of success in a computer science course for non-computer science majors (Stager-Snow, 1985) and found to correlate with grades from a computer science logic course (Kim, 1995). The PLT has been an instrument used in other previous related studies to study Formal Operations thinking (Enyeart, 1981; Platt, 1990; Priebe, 1997; White, 2002; White & Ploeger, 2004; White & Sivitanides, 2005). See Appendix A for PLT examples.

The reliability coefficients of the PLT are .94 and .82 for high school students (Platt, 1990), and .90 and .85 for college students (Enyeart, 1981; Kim, 1995). The validity coefficient was .63 with logical reasoning as measured by the Test of Logical Reasoning (TOLT) (Piburn, 1989). The TOLT is based on Piaget's theory.

3.3 Subjects

Students were from two sections of an introductory programming course in Visual Basic v6 from a central Texas university. The Visual Basic course covered visual objects, controls, events, data types, and procedures. Procedures included logical operations, repetition, and arrays. The students wrote 6 programming projects. The only required competency for this programming course was a computer literacy course dealing with work processing, spreadsheets, and web browsers.

The researcher provided PLT and release/survey forms to 87 college students in a first programming course in Visual Basic v6. These students were exposed to the same course content, instructor, and test. These intervening variables were kept constant in an effort to reduce statistical error variance. This ensured consistent test content and presentation of material. Participation was voluntary and anonymous. The data collected were Pre and Post PLT scores of 0 to 16.

3.4 Data collection and recording

At the beginning of the semester the researcher passed out a PLT sheet and release/survey forms to college students in a first programming course in Visual Basic v6 at a university. The data was obtained from those who wished to participate. At the end of the course the researcher again passed out a PLT sheet to obtain the post-treatment scores. Of the 87 students who started, 51 completed the post-PLT instrument.

4. RESULTS

The SPSS package was used for data analysis. A t-test and paired sampled correlation were performed on the 51 pairs of Pre and Post PLT scores. Table 1 shows that there was no significant difference between the Pre-test and Post-test. Table 2 indicates responses were consistent between the administrations of the PLT. The null hypothesis is tenable.

A comparison was performed on the pre-PLT score between those who completed the study and those that did not (Table 3). A t-Test, assuming unequal variances, almost reached significance ($t = 1.960$, $p < .0536$, two-tail test). There was a tendency for those with low pre-PLT scores to drop the course or not participate with the post-PLT.

5. DISCUSSION

Previous studies showed relationships between cognitive development and success in a programming course. Cause and effect have never been shown between these two variables. This study shows no effect by a visual programming course on cognitive development.

	Pre Mean	SD	Post Mean	SD	t	df	Sig.
PLT	9.7059	4.2251	10.1961	4.7582	-0.870	50	p < .388

Table 1. Score Means, Standard Deviations, Paired Sample T-Test, N = 51, (2-TAIL)

Pairs	Correlation	Sig
Pre & Post PLT	.605	.000

Table 2. Paired Samples Correlations N = 51

Participated Mean	Dropped Mean	t	df	Sig. (2-tail)
9.70	7.89	1.960	75	p< .0536

Table 3 Pre-Plt Score Means Between Those Completing Study And Those That Did Not, T-Test Assuming Unequal Variances, N = 51 & 36

5.1 Limitation of Study

Students who had low cognitive development may have either not participated or dropped out of the course. These students would have been most sensitive to an impact. The students who finished the course may already have been at a high level of cognitive development. This study shows no impact on those who completed the course; not those who dropped out before the end of the course. Those that dropped out or chose not to participate may have done so for many reasons; enrolled in too many courses, work, poor study habits, and a lack of interest/motivation. A presumption to this study is that if the course did have a positive impact on cognitive development, the students would have most likely continued the course and participated. Another presumption was that there was room for improvement. The mean PLT pre-test score was 9.7 out of a possible 16.

The length of treatment was only one semester course of 3 hours of class per week for 15 weeks. There is the possibility that improvement may occur after years of constant treatment. However, if full maturation occurred, it is unlikely there will be a significant improvement.

A few students may not have reached maturation and that the course did effect their cognitive development. Such an outcome maybe hidden from the results since a large majority of the students may have reached maturation. Further study is warranted to see if a programming course effects the cognitive development of a small minority of students who may not have yet reach cognitive maturation.

The sample size was 51 subjects. If the sample size is larger, the analysis would be more sensitive to any change. Future studies of programming impact should consider a larger sample and a treatment of over years. For example, pre-test entering college Freshman majoring in computer programming. Then post-test them in their Senior year. Such a finding would show that, although cognitive maturation may have occurred, cognitive development can improve for those over 18. Unfortunately, those with lower cognitive development level will tend to drop out of the programming curriculum.

6. CONCLUSION

Research has shown that programming can teach cognitive skills but not improve cognitive development, especially for those past the age 18. Studies using adults and students from universities tend not to show improvement since cognitive development probably reached maturation.

Students who have the cognitive ability found at Piaget's Formal Operations are the ones who succeed in a programming course. Those at a lower level of cognitive development tend to drop out of the programming course.

White & Sivitanides (2002) theory suggests that cognitive development can follow a progression from visual through procedural and object oriented programming paradigms. The notion is that visual objects, manipulated on the video display screen, are more concrete while objects used in C++ and Java are more abstract. Because visual items are concrete representations, it follows that pre-Formal Operational Piagetian thinkers will benefit from learning Visual Basic. However, this study shows that visual programming does not improve cognitive development for adults and college students as measured by the PLT.

This study, as well as other studies, indicates students need to be at the required cognitive level in order to succeed in programming. The implication is that programming courses need prerequisites to ensure that students are at the needed cognitive level.

7. REFERENCES

Azzedine, A. (1987). "The relationship of cognitive development, cognitive style and experience to performance on selected computer programming tasks: An exploration." Dissertation Abstracts, Vol. B48, No. 6, p. 1799.

Barker, P. M. (1985). "Cognitive Correlates of Performance in Computer Programming by Children and Adolescents." Dissertation Abstracts, Vol. 46, No. 7, p. 1872.

Bastian, S., Frees, J., Gruber, L., Johnson, J., Landes, B., Morton, L., Rozgony, S., & Stewart, J. (1973). "Are I.S.U. Freshman Students Operating at a Formal level of thought Processes?" Contemporary Education, Vol. 44, pp. 358-362.

Becker, H. J. (1982). "Microcomputers in the Classroom -- Dreams or realities?" ERIC(ED217872).

Bernardo, M. & Morris, J. (Summer, 1994). "Transfer effects of a high school computer programming course on mathematical modeling, procedural comprehension, and verbal problem solution." Journal of Research on Computing in Education, Vol. 26, No. 4, pp. 523-536.

Biehler, R. F. & Snowman, J. (1986). Psychology Applied to Teaching. Houghton Mifflin Company, Boston, MA. p. 62-64

Brainard, C. J. (1978). Piaget's Theory of Intelligence. Englewood Cliffs, NJ: Prentice-Hall.

Cafolla, R. (1987). "The Relationship of Piagetian formal Operations and other cognitive factors to computer programming ability (Development)". Dissertations Abstracts, Vol. A47, No. 7, p. 2506.

Chiapetta, E. (1976). "A review of Piagetian studies relevant to science instruction at the secondary and college level." Science Education, Vol. 60, pp. 253-261.

Choi, W.S. & Repman, J. (1993). "Effects of Pascal & FORTRAN programming on problem-solving abilities of college students." Journal of Research on Computing in Education, Vol. 25 No. 3, pp. 290-303.

Clements D. H. & Gullo, D. F. (1984). "Effects of computer programming on young children's cognition." Journal of Educational Psychology, Vol. 76, No. 6, pp. 1051-1058.

Dalbey, J., & Linn, M. C. (1985). "The Demands and Requirements of Computer Programming: A Literature Review." Journal of Educational Computing Research, Vol. 1, No. 3, pp. 253-74.

Enyeart, M. A. (1981). "Relationships among propositional logic, analytical reasoning, and Piagetian level." Dissertation Abstracts International, Vol. A41, No. 09, p. 397.

Epstein, H (1980). "Some biological bases of cognitive development." Bulletin of the Orton Society, Vol. 30, pp. 46-52.

Epstein, H. (1990). "Stages in human mental growth." Journal of Educational Psychology, Vol. 82, pp. 876-80.

Evans, G. E., & Simkin, M. G. (Nov 1989). "What best predicts computer proficiency?" Communications of the ACM, Vol. 32, No. 11, pp. 1322-1327.

Fischer, K. W., & Silvern, L. (1985). "Stages and individual differences in cognitive development." Annual Review of Psychology, Vol. 36, pp. 613-648.

Fletcher, S. H. (1984). "Cognitive Abilities and Computer Programming." EDRS(ED259700).

Flores, A. (1985). "Effects of computer programming on the learning of calculus concepts." Dissertation Abstracts, Vol. 46, No. 12A, p. 3640.

Folk, M. J. (1973). "Influences of Developmental Level on a Child's Ability to Learn Concepts of Computer Programming." Dissertation Abstracts International, Vol. 34, No. 3, p. 1125a.

Galton, A. (1992). "Logic as a Formal Method." The Computer Journal, Vol. 35, No. 5, pp. 431-440.

Gibbons, P. (1995). "A cognitive processing account of individual differences in novice Logo programmers' conceptualization and use of recursion." Journal of Educatinoal Computing Research, Vol. 13, No. 3, pp. 211-226.

Gibbs, H., & Tucker, A. B. (1986). "A Model Curriculum for a Liberal Arts degree in Computer Science." Communications of the ACM, Vol. 29, No. 3, pp. 202-210.

Griffiths, D. H. (1973). "The Study of the Cognitive Development of Science Students in Introductory Level Courses." ERIC(ED096108).

Hart, M. (1983). "Computer Programming and Algebra. The Nottingham programming in mathematics project." ERIC Report No. ED 227 002.

Hudak, M. A. & Anderson, D. E. (1990) "Formal Operations and Learning Style Predict Success in Statistics and Computer Science Courses." Teaching of Psychology, Vol. 4, pp. 231-234.

Ignatuk, N. (1986). "An Analysis of the Effects of Computer Programming on Analytical and Mathematical skills of high school students." Dissertation Abstracts, Vol. A47, No. 3, p. 854.

Inhelder, B., & Piaget, J. (1958). The Growth of Logical Thinking from Childhood to Adolescence. New York: Basic Books.

Irons, D. M. (1982). "Predicting Programming Performance in Novice Programmers by Measures of Cognitive Abilities." Dissertation Abstracts, Vol. B43, No. 4, p. 1283.

Kim, Y. (1995). "The Reasoning Ability and Achievement of College Level Students enrolled in a Logic Class in Computer Science." University of Texas, Austin.

Kraft, R. H. (1976). "An EEG Study: Hemispheric Brain Functioning of Six to Eight Year Old Children During Piagetian and Curriculum Tasks with Variation in Presentation Mode." ERIC, ER137070.

Kuhn, d. and Langer, J. and Kohlberg, L. and Haan, N. (1977). "The development of formal operations in logical and moral judgment." Genetic Psychology Monographs Vol. 95, No. 1, pp. 97-188.

Little, L. F. (1984). "The Influence of Structured Programming, Gender, Cognitive Development and Engagement on the Computer Programming Achievement and Logical Thinking Skills of Secondary Students." Dissertation Abstracts, Vol. A45, No. 6, p. 1708.

Lovell, K. (1961). "A follow-up study of Inhelder and Piaget's the growth of logical thinking." British Journal of Psychology, Vol. 52, pp. 143-153.

Mains, M. G. (1997). "The Effects of Learning a Programming Language on Logical Thinking Skills." Unpublished Thesis, University of Nevada, Las Vegas, Nevada.

Many, W. A. and Lockard, J. and Abrams, P. D. and Friker, W. (1988). "The effect of learning to program in Logo on reasoning skills of Junior High School students." Journal of Educational Computing Research, Vol. 4, No. 2, pp. 203-213.

McCoy, L. P. (1987). "The effect of computer programming experience on mathematical problem-solving ability." Dissertation Abstracts, Vol. 48, No. 05A, p. 1105.

Monfort, M., Martin, S. A., & Frederickson, W. (1990). "Information-processing differences and laterality of students from different colleges and disciplines." Perceptual & Motor Skills, Vol. 70, No. 1, pp. 163-172.

Myers, J., J. P. (1990). "The central role of mathematical logic in computer science." SIGCSE Bulletin, Vol. 22, No. 1, pp. 22-26.

Ott, C. F. P. (1989). "Predicting achievement in computer science through selected academic, cognitive and demographic variables." Dissertation Abstracts, Vol. A49, No. 10, p. 2988.

Owens, B. and Seiler, J. (1996). "A study of the relationship between performance in propositional logic and computer science." Proceedings of the Seventh Annual south Central Small College Computing Conference, The Journal of Computing in Small Colleges, Vol. 11, No. 7, pp.90-93.

Piaget, J. (1972). "Intellectual evolution from adolescence to adult." Human Development, Vol. 15, pp. 1-12.

Palumbo, D. & Reed, M. (Spring, 1991). "Effect of BASIC programming language instruction on high school students' problem solving ability and computer anxiety." Journal of Research on Computing Education, Vol. 23, No. 3, pp. 343-372.

Pea, R. D. & Kurland, D. M., (1984). "On the cognitive effects of learning computer programming: a critical look." Technical Report No. 9, Bank Street College of Education, New York.

Petrushka, D. (1984). "A Study of the effect of content on the ability to do syllogistic reasoning: An investigation of transferability and the effect of practice." Unpublished doctoral dissertation, Rutgers University, NJ.

Piburn, M. D. (1989). "Reliability and validity of the Propositional Logic Test." Educational and Psychological Measurement, Vol. 49, pp. 667-72.

Platt, D. C. (1990). "The effects of a second-semester computer programming course on mathematical problem-solving performance and the relationship between selected cognitive factors and mathematical program-solving performance." Dissertation Abstracts, Vol. 51, No. 10A, p. 3354

Priebe, Roger L. (1997). "The Effects of Cooperative Learning on Content Comprehension and Logical Reasoning in a Second-Semester University Computer Science Course." Unpublished Doctoral Dissertation, University of Texas, Austin, Texas.

Renner, J., Grand, R., & Sutherland, J. (1978). "Content and concrete thought." Science Education, Vol. 62, pp. 215-221.

Ricardo, C. M. (1983). "Identifying student entering characteristics desirable for a first course in computer programming." Dissertation Abstracts, Vol. A44, No. 1, p. 96.

Ross, R. (1982). "Learning Spurts and Plateaus." Educational Resource Information Center (ERIC), ED234035, 48-52.

Schwebel, M. (1972). "Logical Thinking in College Freshman: Final Report." ERIC(ED 110896).

Schwebel, M. (1975). "Formal operations in first year college students." Journal of Psychology, Vol. 91, pp. 133-141.

Seidman, R. H. (1989-90). "Computer Programming and Logical Reasoning: Unintended Cognitive Effects." Journal of Educational Technology Systems, Vol. 18, No. 2, pp.123-141.

Sein, M. K., & Bostrom, R. P. (1989). "Individual differences and conceptual models in training novices users." Human-Computer Interaction, Vol. 4, pp. 197-229.

Shaw, D. G. (1984). "The effects of learning to program a computer in BASIC or Logo on the problem solving abilities of fifth grade students." Dissertation Abstracts, Vol. A45, No. 7, p. 1985.

Soloway, E. (October, 1993). "Should we teach students to program?" Communications of the ACM, Vol. 36, No. 10, pp. 21-24.

Sperschneider, V., & Antoniou, G. (1991). Logic: A foundation for Computer Science. Reading, MA: Addison-Wesley.

Stager-Snow, D. B. (1985). "Analytical ability, logical reasoning and attitude as predictors of success in an introductory course in computer science for non-computer science majors." Dissertation Abstracts, Vol. A45, No. 8, p. 2473.

Stockwell, W. F. (2002). "The effects of learning "C" programming on college students' mathematics skill." Dissertation, University of Oklahoma, Norman, Oklahoma.

Sund, R. B. (1976). "Piaget for educators: a multimedia program" (In Brooks, R (1978) The relationship between Piagetian Cognitive development and cerebral cognitive asymmetry. ERIC ED160224, Trans.). Columbus: Charles E. Merrill.

Wu, C.-C. (1993). "Conceptual Models and Individual Cognitive learning Styles in Teaching Recursion to Novices." Unpublished Dissertation, University of Texas, Austin.

White, G. (Spring, 2002). "Cognitive Characteristics for Learning C++." Journal of Computer Information Systems, Vol. 42, No. 3, pp. 51-55.

White, G. and Ploeger, F. (Spring, 2004). "Cognitive Characteristics for Learning Visual Basic." Journal of Computer Information Systems , Vol. 44, No. 3, pp. 58-66.

White, G. and Sivitanides M. (2002). "A Theory of the Relationships between Cognitive Requirements of Computer Programming Languages and Programmers' Cognitive Characteristics." Journal of Information Systems Education, Vol. 13, No. 1, pp. 59-66.

White, G. and Sivitanides, M. (2005). "Cognitive Differences between Procedural and Object Oriented Programming." Journal of Information Technology and Management, Vol. 6, No. 4, pp. 333-350.

AUTHOR BIOGRAPHY

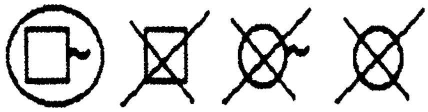
Garry L. White is a faculty member in the Computer Information Systems department at Texas State University in San Marcos Texas. He holds a MS in Computer Sciences from Texas A & M University – Corpus Christi and a PhD in Science Education, emphasis in Information Systems, from The University of Texas at Austin. Professional Certifications from the Institute of Certified Computer Professionals (ICCP) include C.D.P, C.C.P., and C.S.P. He has been on the Texas State faculty since 1997. His teaching interests are in the areas of Computer Programming, Data Communications, Systems Analysis, and Computer Networks. His research interests and work are in the areas of Computer Education and the Internet. He has published papers and abstracts in journals such as the Journal of Computer Information Systems. Proceeding publications have been with the Decision Sciences Institute and the Information Systems Educational Conference.

APPENDIX A

Propositional Logic Test (PLT)
Instructions

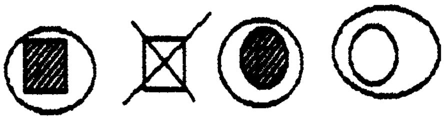
In each of the problems on the PLT you will find a sentence followed by four figures. Each figure is either square or round, either large or small, either white or striped, and either tailed (has a tail) or untailed. Your task is to circle those figures that are allowed by the sentence and to cross out the ones that are not allowed. Here are some examples with the correct answers to show what this means. Study them carefully since the problems that follow are very similar.

Ex 1. It is square and it is tailed.



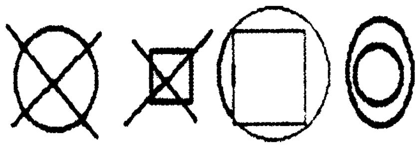
Here it says it must be square and tailed so only the one that is both square and tailed fits. The others are not tailed or are not square or are not both so that they should be crossed out.

Ex 2. If it is white then it is round.



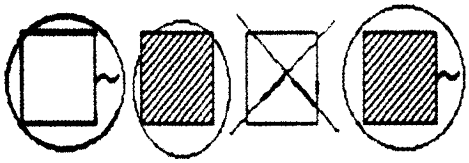
If it is White then it must be round, but if it is striped then it doesn't matter if it's round or not. So the white circle fits but the white square does not. The striped figures all fit because the statement only tells us about white figures.

Ex 3. If it is round it is small and if it is small it is round.



The round ones that are small fit and so do the small ones that are round. Since the large square isn't round it doesn't have to be small, so it fits. The large circle doesn't fit the first part of the rule and the small square doesn't fit the second part.

Ex 4. It is striped or it is tailed or both.



You can circle the first figure because it is tailed. The second figure also fits because it is striped, and the last one fits because it is both striped and tailed. The third figure doesn't fit since it is neither striped nor tailed.

You have 15 minutes to complete the test.

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096