A Theory of the Relationships between Cognitive Requirements of Computer Programming Languages and Programmers' Cognitive Characteristics

Garry L. White, Ph. D.
GW06@business.swt.edu

Marcos P. Sivitanides, Ph.D.
Department of Computer Information Systems
College of Business Administration
Southwest Texas State University
San Marcos, TX 78666

ABSTRACT

This paper formulates a theory that investigates the possible effects of two human cognitive characteristics, on the difficulties of learning specific programming languages. The two human cognitive characteristics are Piaget's cognitive development and McCarthy's cognitive hemispheric style. This paper consolidates prior research and accepted cognitive theory. It then presents a formulation of a theory that relates cognitive requirements of different computer programming languages and programmers' cognitive characteristics. If the cognitive requirements for a programming language are beyond the cognitive characteristics of a programming student, the student may burn out. If the cognitive requirements are below the student's cognitive characteristics the student may be bored. If they are similar to them, the student is able to meet the challenges. Motivation, interest, self-esteem and success may thus be optimized. Different programming languages are more suited for different cognitive characteristics. This theory extends prior research in cognitive theory and cognitive requirements of computer programming.

Keywords: Cognitive development, Cognitive style, Programming languages, Script Programming, Procedural programming, Object-oriented programming, Visual programming

1. INTRODUCTION

"There is a need to understand how people learn, not just aptitude. Such understanding may influence productivity in various programming languages" (Myers 1996). Research is needed to improve understanding of the learning process and identify the underling cause of students' difficulties with programming languages. "Study of the language-learning process is necessary to understand how the process can be improved" (Myers 1996). A research study with computer science courses emphasized "the need to examine students' cognitive maturity and learning style -- factors often ignored in research aimed at ascertaining the reasons for academic success at the college level." The findings of that study "highlight the need to examine both cognitive maturity and learning style in the studies of academic success at the college level" (Hudak 1990). This type of research can enhance academic teaching and industry training. (Rosson 1990; Scholtz 1993; Sheetz 1997).

Why do some students take computer programming courses and fail, while others succeed? Research has shown that novice college computer science students experience more difficulty with concepts involving mathematical logic, than they do with other concepts (Almstrum 1994). Cafolla (1987) found that "... some people of college age have difficulty in learning procedural programming. This suggests that the cognitive skills needed to learn procedural programming develop later or perhaps never, in some". Is it possible that these students lack the required cognitive characteristics to learn programming? This begs the question: Which hemispherical cognitive style and which stage of cognitive development are better suited for different computer programming language paradigms?

This paper focuses on two human cognitive characteristics: (1) cognitive development and (2) cognitive hemispheric dominance (cognitive style). The different programming language paradigms, whose cognitive requirements are considered in this paper, are: procedural, object-oriented, visual, and script.

It should be noted that the impact of these cognitive factors can vary in strength due to differences in course content. For example, a programming paradigm may involve concepts that favor the left side of the brain, while another one may involve concepts that favor the right side. One programming paradigm may focus on object manipulation, while another may focus on problem solving skills and the flow of logic through the program.

2. COGNITIVE CHARACTERISTICS

Research has shown that cognitive development (what can be learned), cognitive styles (how one learns), and prior experiences are factors in learning procedural programming languages (Losh 1984; Fletcher 1984; Little 1984; Ott 1989; Monfort 1990). Myers (1996) showed that different learning styles were significant predictors of achievement between Imperative (Procedural) and Functional (Non-Procedural) programming methods. Bishop-Clark (1995) found that cognitive style affected programming performance.

2.1 Cognitive Development

Piaget’s cognitive development theory deals with three stages of development (Piaget 1972; Epstein 1990), pre-operational, concrete, and formal operations. Pre-operational cognitive level involves the mental age from age 2 years to age 7 years.

The concrete level person, mental age of 7 years to 12 years, understands conservation of matter and classification/generalization (conclude that all dogs are animals and not all animals are dogs). However, such a person is unable to comprehend mathematical ratios (Barker 1983).

Formal operations is the highest cognitive development level defined by Piaget. It is the ability to deal with abstractions, form hypotheses, solve problems systematically, and engage in mental manipulations.” (Biehler and Snowman 1986). A precondition to formal operations development is to understand biconditional reasoning, “if and only if” logic (Lawson 1983). This is significant for procedural programming. Procedural programming logic uses the biconditional reasoning of “if and only if” logic.

> Children younger than 11 or 12 find it difficult to learn procedural programming (Becker 1982). This suggests there is some type of cognitive development that allows older children above this age range to learn procedural programming. Since procedural programming skills are related to logical reasoning (Folk 1973; Fletcher 1984; Cafolla 1987), it is not surprising that younger children are unable to do programming in light of Piaget's theory of cognitive development.

Piaget's theory fosters the notion that formal operational thinking abilities develop around age 11-12 (Chiapetta 1976). It is at this age that students begin to move from concrete thinking to logic/abstract thinking. Research has shown that these formal operations, such as thinking in abstractions and logically, occur much later in some people or not at all (Griffiths 1973; Schwebel 1975; Pallrand, 1979).

> Research has shown that 17% of 7th graders, 23% of 8th graders, and 34% of 12th graders reach formal operational thinking abilities (Renner 1978). Similar findings were made by Epstein (1980), when he showed that development through Piaget's stages was by degree. For example, while 20% of 13 year olds (8th graders) were at the formal operational stage, 78% were at the concrete operational stage and 2% were at the pre-operational stage of cognitive development.

Several studies show that a majority of adults, including college students and professionals, fail at many formal operational tasks (Sund 1976; Petrushka 1984). Many college students fail to attain full formal operational thinking (Griffiths 1973; Schwebel 1975). There are adults who’s cognitive development is at the concrete level, mental age of 7 years to 12 years.

Different people develop their formal operational thinking abilities at different rates and may reach different maximum levels. Why do so many, never reach the formal level of thinking? The reason has been identified to be dependant upon the maturing neural fibers between the left and right cerebral hemispheres (Kraft 1976). The advancement of people through the development of Piagetian stages is an indication of such maturation. Ross (1982) found that Epstein's descriptions of growth spurts and plateaus corresponded to Piaget’s learning stages.

2.2 Cognitive Style

Different people process the same information in different ways using different areas of the brain, depending upon their cognitive style. Hemisphericity is a term used to describe how the brain processes specific information, and research suggests that one side predominates over the other (Losh 1984). The left brain functions differently from the right brain (Saleh 1995; Supprian 1997). Examples of some left hemispheric characteristics are: talking/writing and rational, objective judgments. Examples of some right hemispheric characteristics are: intuitive, subjective judgment, and drawing/manipulating physical objects (McCarthy 1986).

Electroencephalograms (EEG's) have shown that different cognitive styles use different sides of the brain (Riding 1997). This leads to further hemispheric differences (Gordon 1988), because the right and left cerebral hemispheres process information differently. Which hemispherical cognitive style is best for different computer programming languages? Studies using EEG measurements have shown that cognitive tasks activate different parts of the brain (Jausovec 1997). EEG measurements have shown that the left brain deals with Piagetian tasks of logic (Kraft 1976) and EEG measurement showed increased activity in the left hemisphere when subjects performed arithmetic (Rotenberg 1997).

Geschwind & Galaburda (1985) found many studies showing that each hemisphere is usually superior over the other in certain cognitive functions and that the left hemisphere matures later than the right. The right side of the brain seems to handle concrete experiences and the left side of the brain seems to process abstract conceptions (Diehl 1986). Another study showed that the left brain is the logical cognitive side and that the right brain is the creative cognitive side (Herrmann 1981). Other studies have shown that the left side of the brain also deals with logical cognition (Lawson 1975). A more recent study found some cooperation between the hemispheres involving reasoning. The left brain dealt with probabilistic reasoning and the right brain dealt with deductive reasoning (Osherson 1998).

3. COGNITIVE CHARACTERISTICS AND PROGRAMMING LANGUAGES TYPES

3.1 Procedural Languages

Most programming languages are Procedural. Such a language is "characterized by these three properties: the sequential execution of instructions, the use of variables representing memory locations and the use of assignment to change the values of variables" (Louden 1993). The data is kept separately from the procedures within the same program. An example of such a language is COBOL. The definitions of data used in the program are placed in separate code away from the instruction code.

3.1.1 Cognitive Development

Some research suggests that programming involves important higher cognitive abilities (Hudak 1990) such as problem solving and Piaget's formal operations. Other studies have shown that formal operational reasoning ability is necessary for success in procedural computer programming/logic (Fletcher 1984; Little 1984; Azzedine 1987; Hudak 1990).

Azzedine (1987) tested 203 students from the 6th grade to college level with the Langeot Test of Cognitive Development. This research investigated the implications of Piaget's cognitive developmental theory and the intellectual prerequisite of learning procedural programming. The results showed that cognitive development predicted programming performance.

Cafolla (1987) did a similar study with students from a community college. Each student was given the Inventory of Piaget's Development Tasks (IPDT) to measure his or her cognitive development level. The results were the same as for Azzedine (1987): cognitive development predicted programming performance.

Little (1984) found that students who tested high in formal operations, Piaget's high level of cognition, scored higher on programming and logical thinking measures than those students who were concrete operational thinkers (a Piaget's lower level of cognition). This cognitive developmental level is a factor in determining one's ability to learn procedural programming (Folk 1973). This finding is also supported by Hudak & Anderson (1990). They determined that people who have reached Piaget's formal operational stage, would have the tools needed to understand programming. They also have a greater abstract learning style that helps them learn programming.

3.1.2 Cognitive Style

Students who are successful in procedural programming have been found to be significantly left hemispheric brain dominant for cognitive style (White 2001). This was true at public, post-secondary and vocational-technical schools where "Your Style of Learning and Thinking-Form C" inventory forms were used (Losh 1984). A later study found Computer Science and Mathematics students also to be left brain dominant while music, art, oral communication and journalism students were found to be right brain dominant. Brain hemisphere dominance was inferred from Human Information Processing Survey scores (Monfort 1990).

A dissertation by Ott (1989) supports the above findings. She found that left brain dominance in high school students, correlated significantly with grades in procedural programming courses (r = .30 & .34). Brain dominance was determined by the Herrmann Participant Survey Form.

3.2 OOP Languages

Object oriented programming "is based on the notion of an object, which can be loosely described as a collection of memory locations together with all the operations that can change the values of these memory locations" (Louden 1993). Data declarations, data definitions and program instructions are all under one identifier, known as an object. Examples of this type of paradigm language are C++ and Java.

Most research dealing with the cognitive aspects of programming dealt with Procedural programming languages, such as COBOL, BASIC, Pascal and FORTRAN. There is very little research dealing with cognitive characteristics required for OOP.

3.2.1 Cognitive Development

What is known about OOP indicates that development of a program uses problem solving skills, a high cognitive level (Kim 1997). A recent research study did show that OOP also involved Piaget's formal operational cognitive level (White 2001). More research in this area is warranted.

3.2.2 Cognitive Style

Cognitive style appears to be hemispheric friendly. All hemispheric styles appear to be able to learn OOP (White 2001). This may be due to the fact that user cognition has shown Object Oriented properties of cognitive economy and limited storage space (Krovi 1998). More research in this area is warranted.

3.3 Visual Languages

There is a lack of research describing required cognitive characteristics for Visual Programming. What follows is a formulation of a hypothesized theory based on Piaget's theory and characteristics of the language. Empirical research is warranted, to support or refute this new hypothesized theory.

3.3.1 Cognitive Development

The language characteristic of Visual programming is the manipulation of visual objects on a computer screen. An example is Visual Basic by Microsoft. Some Visual programming languages have OOP and procedural characteristics. Therefore, it is suspected that formal operation cognitive level would be required. However, instead of manipulating abstract objects found in C++ or Java, visual objects on a computer screen are manipulated. Since this is a concrete component, it may be that those who are pre-formal operation thinkers would be able to handle this challenge of visual objects on a screen and be successful. Formal operation thinkers might find it an easy task, since the cognitive characteristic of visual programming has a concrete component. Empirical research that deals with Visual programming and cognitive development is lacking in the literature. Empirical research is warranted to support or reject this hypothesis.

3.3.2 Cognitive Style

Since there are OOP characteristics/concepts with Visual Programming, it is speculated that it would be cognitive (hemispheric) style friendly. Empirical research that deals with Visual programming and cognitive style is lacking in the literature. Empirical research is warranted to support or reject this hypothesis.

3.4 Script Languages

3.4.1 Cognitive Development

What about those who are at a lower level of cognitive development such as concrete operational thinkers as defined by Piaget? A solution might be script programming languages, such as HTML, XML and other web page development languages. Such programming languages develop formats and layouts of visual objects and text on the computer screen. Script programming may be an alternative for those who find procedural programming or OOP difficult. Script languages lack substantial logic and abstract procedures. The user indicates how things are to be displayed on the screen. Instead of using logic and abstract algorithms to query and process data, English like statements could be used to tell the computer what is to be done. Empirical research that deals with Script programming and cognitive development is lacking in the literature. Empirical research is warranted to support or reject this hypothesis.

3.4.2 Cognitive Style

Since the right side of the brain seems to handle concrete experiences and creativity while the left side of the brain seems to process abstract and logic conceptions (Diehl 1986; Herrmann 1981; Lawson 1975), it is hypothesized that Script programming is right hemispheric cognitive style.

However, subjects with mixed hemispheric dominance, based on eye-hand preference, have shown low performance when using HyperCard software. The subjects who were more symmetrical in laterality, left hand-left eye or right hand-right eye, exhibited better performance when designing a sales presentation using HyperCard software (McCluskey 1997).

Table 1. Programming Languages and Cognitive Development/Style

Programming Pardigm	Piaget's Cognitive Development Levels Pre-Oper	Concrete	Pre-Formal	Formal	Cognitive Style (Hemisphericity)
Procedural (COBOL, logic sequence)	Burnout	Burnout	Burnout	P & M	Left Brain
Object Oriented (C++, Java, concepts)	Burnout	Burnout	Burnout	P & M	Either Hemisphere
Visual (Visual Basic, on screen)	Burnout	Burnout	P & M	Bored	Either Hemisphere
Script (HTML, Web Pages)	Burnout	P & M	Bored	Bored	Right Brain

P & M: Productive and Motivated

Empirical research that deals with Script programming and cognitive style based on hemispheric dominance is lacking in the literature. Empirical research is warranted to support or reject this hypothesis.

4. SUMMARY

The literature has shown that formal operational cognitive development is a required cognitive characteristic of people for learning procedural programming. The majority of adults and many college students fail to develop to full formal operational thinking skills. Research has also supported logical thinking skills (a component of formal operational cognitive development) as a required characteristic for learning procedural programming.

Research has shown that procedural and object oriented programming, require the cognitive characteristic of formal operations. Those at this cognitive level would be "productive and motivated" (P & M), able to handle the challenge of procedure programming and OOP. They would have the mental tools to be successful. Table 1 shows a conjecture that those students who are below this cognitive level would "burnout" in such a programming class. The required cognitive characteristic of the language is beyond the cognitive development of the student.

The literature has shown left hemispheric thinking style of learners as another characteristic necessary for success with procedural programming. Since schools tend to teach to the left hemispheric thinking style (Hatcher 1983; Walden 1995), this may explain why many right brain thinkers have problems with programming courses taught in schools.

5. EDUCATIONAL IMPLICATIONS

If a teacher uses subject material that caters to the left side of the brain, right dominant brain students will have trouble (Creswell 1988). If the content level exceeds the cognitive level of the students, the students will burnout. There is the risk that the students' self-esteem will be damaged. As shown in Table 1, if the students' cognitive level exceeds the course content level, the students will be bored. The students' interest and motivation will be hindered.

It is recognized that individuals learn differently and have different instructional needs (Sonnier 1976). To be most effective, teaching styles and content level must be compatible with the cognitive development and style of an individual. It is beneficial to the students that computer programming courses have prerequisites that place them in a course that best fits their cognitive characteristics. Motivation, interest, self-esteem, and success may thus be optimized.

A way to implement some type of prerequisites, is to use standardize math scores from the ACT and SAT. The research literature supports the relationship between mathematic scores and success with procedural programming languages (Ricardo 1983; Ignatuk 1986; Renk 1987; Ott 1989). If the learner is weak in mathematics, the placement would be with Script or Visual programming. If the learner is strong in mathematics, the placement would be with procedural or OOP programming. Again, research to show relationships between mathematic scores and success with Script, Visual, and OOP programming languages is lacking in the literature.

Another avenue for prerequisites is for the learner to start with Visual programming such as Visual Basic. If the learner does poorly, the next course would be a Script language, such as HTML or XML. If the learner does well in Visual programming, the next course would be a procedural or OOP language, such as Java or C++.

6. POTENTIAL FOR FUTURE RESEARCH AND CONCLUSIONS

Some programming classes may have a bimodal distribution of students' grades. The low mode may indicate Piaget's concrete operation stage. The high mode may indicate Piaget's formal operation stage. This is supported by Hudak (1990). That study showed formal operation level students did better then concrete level students in a Statistics course and an Introduction Computer Science course. Research in this area is warranted to determine if bimodal distribution of grades in OOP, Visual, and Script courses are differentiating between concrete and formal operation thinkers.

In conclusion, when we are able to ascertain the cognitive characteristics of students and place them in courses that require similar cognitive characteristics, we can expect a high level of success from the students in the class. Furthermore, if we can show that Script and Visual programming languages can improve cognitive development or change cognitive style, we can use these programming language courses to better prepare those students who lack the required cognitive characteristics to be successful in procedural programming and OOP courses.

7. REFERENCES

Almstrum, V. L. [1994], "Limitations in the Understanding of mathematical Logic by Novice Computer Science Students." Unpublished Dissertation, University of Texas, Austin, Tx.

Azzedine, A. [1987], “The relationship of cognitive development, cognitive style and experience to performance on selected computer programming tasks: An exploration.” Dissertation Abstracts, B48(6), 1799.

Barker, R. J. and E. A. Unger [1983], “A Predictor for Success in an Introductory Programming Class based upon Abstract Reasoning Development.” Proceedings of the 14th SIGCSE Technical Symposium on Computer Science Education of the ACM, Orlando, Florida.

Becker, H. J. [1982], “Microcomputers in the Classroom -- Dreams or realities?” ERIC(ED217872).

Biehler, R. F. and J. Snowman [1986], Psychology Applied to Teaching. Houghton Miffin Compay, Boston.

Bishop-Clark, C. [1995], “Cognitive Style and Its Effect on the Sategs of Programming.” _Journal of Research in Computing Education, 27(4) p373-86.

Cafolla, R. [1987], “The Relationship of Piagetian formal Operations and other cognitive factors to computer programming ability (Development).” Dissertations Abstracts, A47(7), 2506.

Chiapetta, E. [1976], “A review of Piagetian studies relevant to science instruction at the secondary and college level.” Science Education, 60, 253-261.

Creswell, J. L. and C. Gifford and D. Huffman [1988], “Implications of Right/Left Brain Research for Mathematics Educators.” School Science and mathematics, 88(2), 118-131.

Diehl, H. L. [1986 Nov], “Learning Style and Brain Hemisphere Dominance: Interrelationships and Influences on Organizational Role Selection.” Dissertation Abstracts, A50(5), 1363.

Epstein, H. [1980], “Some biological bases of cognitive development.” Bulletin of the Orton Society, 30, 46-52.

Epstein, H. [1990], “Stages in human mental growth.” Journal of Educational Psychology, 82, 876-80

Fletcher, S. H. [1984], “Cognitive Abilities and Computer Programming.” EDRS(ED259700).

Folk, M. J. [1973], “Influences of Developmental Level on a Child's Ability to Learn Concepts of Computer Programming.” Dissertation Abstracts International, 34(3), 1125a.

Geschwind, N. and A. M. Galaburda, [1985], “Cerebral lateralization. Biological mechanisms, associations and pathology. I,II,III: A hypothesis and program of research.” Archives of Neurology, 42, 428-459, 521-552, 634-654.

Gordon, H. W. [1988], “Specialized cognitive function and school achievement.” Developmental Neuropsychology, 4(3), 239-257.

Griffiths, D. H. [1973], “The Study of the Cognitive Development of Science Students in Introductory Level Courses.” ERIC(ED096108].

Hatcher, M. [1983], “Whole Brain learning.” School Administrator, 40(5), 8-11.

Herrmannn, N. [1981], “The Creative Brain.” Training & Development Journal, 35(10), 10-16.

Hudak, M. A. and D. E. Anderson [1990], “Formal Operations and Learning Style Predict Success in Statistics and Computer Science Courses.” Teaching of Psychology, 17(4) 231-234.

Ignatuk, N. [1986], “An Analysis of the Effects of Computer Programming on Analytical and Mathematical skills of high school students.” Dissertation Abstracts, A47(3), 854.

Jausovec, N. [1997], “Differences in EEG Alpha

Activity between Gift and Non-Identified Individuals: Insights into Problem Solving." Gifted Child Quarterly, 41(1), 26-32.

Kim, J. and F. J. Lerch, [1997], "Why is programming so difficult? Programming as scientific discovery in multiple problem spaces." Information Systems Research_8(1), 25-50.

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).

Krovi, R. and A. Chandra [1998], "User cognitive characteristics: The case for an object oriented model." Journal of Systems & Software. 43(3), 165-176.

Lawson, A. E., and W. T. Wollman [1975]. "Hemispheric Dominance, Conservation Reasoning and the Dominant Eye." EDRS, ED130778.

Lawson, A. E. [1983], "The Acquisition of Formal Operational Schemata during Adolescence: The Role of the Biconditional." Journal of Research in Science Teaching, 20(4), 347-56.

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, A45(6), 1708.

Losh, C. L. [1984], "The relationship of student hemisphericity to performance in computer programming courses." Dissertation Abstracts A44(7), 2127.

Louden, K. C. [1993], Programming Languages, Principles and Practice. PWS Publishing Company, Boston.

Monfort, M. and S. A. Martin and W. Frederickson [1990], "Information-processing differences and laterality of students from different colleges and disciplines." Perceptual & Motor Skills, 70(1), 163-172.

Myers, J. P. & Brita , M. [1996]. "Cognitive Style and Achievement in Imperative and Functional Programming Language Courses." ERIC ED398892 (1996).

Osherson, D. and D. Perani, and S. Cappa and T. Schnur and F. Grassi and F. Fazio [1998], "Distinct Brain Loci in Deductive versus Probabilistic Reasoning." Neuropsychologia, 36(4), 396-376.

Ott, C. F. P. [1989], "Predicting achievement in computer science through selected academic, cognitive and deographic variables." Dissertation Abstracts, A49(10), 2988.

Paget, J. [1972]. Intellectual evolution from adolescence to adult. Human Development, 15, 1-12.

Pallrand, G.J. [1979], "The transition to formal thought." Journal of Research in Science Teaching, 16, 445-451.

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.

Renk, S. C. [1987], "Factors Affecting Academic Success in Introductory Computer Programming." Dissertation Abstracts, A48(3), 579.

Renner, J. and R. Grand and J. Sutherland [1978], "Content and concrete thought." Science Education, 62, 215-221.

Ricardo, C. M. [1983], "Identifying student entering characteristics desirable for a first course in computer programming." Dissertation Abstracts, A44(1), 96.

Riding, R. J. and A. Glass and S. R. Butler and C. W. Pleydell-Pearce [1997], "Cognitive style and individual differences in EEG alpha during information processing." Educational Psychology, 17(1-2), 219-234.

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

Rosson, M.B and J. M. Carrol and R. K. E. Bellamy [1990], "A case study in minimalist instruction." Human Factors in Computing Systems, CHI '90 Converence, p. 423-429.

Rotenberg, V. S. and V. V. Arshavsky [1997], "Right and left brain hemispheres activation in the representatives of two different cultures." Homeostasis in Health & Disease, 38(2), 49-57.

Saleh, , A. and A. Iran-Nejad [1995], "Wholetheme constructivism and Whole Brain Education: Educatinoal Implications of the Research on Left and Right Brain Hemispheres." ERIC(ED393896).

Scholtz, J. and S. Chidamber and R. Glass and A. Goemer and M. B. Rosson and M. Stark and I. Vessey [1993], "Object-oriented programming: the promise and the reality." Journal of Systems Software, 23, 199-204.

Schwebel, M. [1975], "Formal operations in first year college students." Journal of Psychology, 91, 133-141.

Sheetz, S. and G. Irwin and D. Tegarden and H. J. Nelson and D. E. Monarchhi [1997], "Exploring the difficulties of learning object-oriented technique." Journal of management Information Systems. 14(2) 103-131.

Sonnier, I. L. [1976], "Logic Patterns and Individual Differences." Southern Journal of Educational Research, 10(3), 136-150.

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.

Supprian, T. and E. Hofmann [1997], "The fornix of the human brain: Evidence of left/right asymmetry on axial MRI scans." Surgical and Radiologic Anatomy, 19(2), 105.

Walden, U. [1995], "The neuropsychological implications of hemispheric learning and teaching preferences on the achievement of college prep students: In search of brain based education." Dissertation Abstracts, A55(9), 2780

White, G. L. [2001], "Cognitive Characteristics for Learning Java, an Object Oriented Programming Language." Unpublished Dissertation, University of Texas at Austin, Texas

AUTHOR BIOGRAPHIES

Garry L. White is a faculty member in the Computer Information Systems depart-ment at Southwest Texas State University (SWT) 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 SWT 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.

Marcos P. Sivitanides is a tenured Associate Professor of Computer Information Systems at Southwest Texas State University (SWT) in San Marcos Texas. He holds a BA (Honors) in Computer Sciences, an MBA and a PhD in Management Information Systems, all from The University of Texas at Austin. He has been on the SWT faculty since 1989. His teaching interests are in the areas of Computer Programming, Systems Analysis and Design, Database Design and Management, and Computer Networks. His research interests and work are in the areas of Decision Theory, Computer Education, and Curriculum Development and Design. He has published papers and abstracts in journals such as Decision Sciences and the Journal of Information Systems Education and conferences such as the Decision Sciences Institute, and the Information Systems Educational Conference.

Information Systems & Computing
Academic Professionals

EDSIG
Serving Information Systems Educators

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