# Chapter 3.32
# Assessing the Effectiveness of Programmed Instruction and Collaborative Peer Tutoring in Teaching Java™

**Henry H. Emurian**
*University of Maryland, Baltimore County (UMBC), USA*

## ABSTRACT

Students in two Java programming classes completed an individualized tutoring system that taught a simple applet program. Before and after using the tutor, students completed questionnaires that assessed software self-efficacy and understanding of general programming principles. The questionnaires also were administered following a lecture session on the program that included having the students successfully run the applet in a browser on the Web. For the second class, a collaborative peer tutoring session based on the applet program occurred between completion of the tutor and the lecture session. Students in both classes increased in software self-efficacy and program understanding across the assessment occasions. For students in the second class, correct answers on the final test of understanding were higher than observed in the first class. Collaborative peer tutoring used in combination with a programmed instruction tutoring system may potentiate learning for novitiate students.

## INTRODUCTION

Teaching computer programming is not easy (Traynor & Gibson, 2004). Reviews of the instructional literature indicate that many students struggle with their programming courses (Robins, Rountree, & Rountree, 2003), and the complexity and instability of Java pose unique challenges to both educators and students (Roberts, 2004). The research reported here, then, reflects an attempt to improve the effectiveness of Java programming instruction, as evidenced by enhanced student performance, which was accomplished by combining an individualized tutoring system with collaborative peer tutoring. This tactic of teach-

ing students with novel instructional approaches is fundamental to success-oriented classroom strategies, which are mindful of the alarming dropout rates reported for students in many science, technology, engineering, and mathematics (STEM) programs of study (Wormley, 2003).

## Tutoring System

The research group previously reported a series of evaluations in the development of a Web-based tutoring system[1] and its classroom application as the first technical training exercise for students in a Java computer programming course (Emurian, 2004, 2005, in press; Emurian & Durham, 2001, 2002, 2003; Emurian, Hu, Wang, & Durham, 2000; Emurian, Wang, & Durham, 2003). The tutor teaches a simple 32-item, 10-line Java applet that displays a Label object in a browser window on the Web.[2] The learning theory supporting the development of the tutoring system is a behavior analysis model based upon the learn unit formulation (Greer & McDonough, 1999; Singer-Dudek & Greer, 2005) as applied to programmed instruction for technology education (Greer, 2002).

The objective of the tutor is to provide each and every student with elementary knowledge and skill in preparation for continuing study of the Java programming language. The tutoring system is effective in promoting skill and cultivating self-confidence in beginning students by giving them a successful learning experience that motivates their further study of Java using textbooks, lectures, laboratory demonstrations, independent problem solving, and the like. The tutoring system is intended to meet the needs of Information Systems majors, whose professional interests typically are outside of the scope of computer programming.

## Dyadic Collaboration

Although group discussions involving three or more participants may have value in facilitating learning to write computer programs (Davy & Jenkins, 1999), having students study together in dyads recently has been investigated as an even more powerful tactic to improve learning at the level of individuals. There are several variations to the structure of a dyadic interaction, which include reciprocal peer tutoring (Griffin & Griffin, 1998), pair programming (Williams et al., 2000), and interteaching (Boyce & Hineline, 2002). The variant of collaborative peer tutoring that was adopted in the present study is the interteaching dialog, which is a mutually probing, mutually informing conversation between two people (Boyce & Hineline, 2002). Interteaching has the objective of insuring by the participants as a team that each member of a dyad can answer a previously disclosed set of questions. This approach is similar to the peer collaboration paradigm to teach recursion that was studied by Jehng (1997). It is suggested here that an interteaching session following individualized tutoring can potentiate the prior learning and result in enhanced competency and understanding.

## Background and Rationale

Our previous research (Emurian, 2005, in press) showed that students who completed the Java tutoring system learned general rules of Java programming that could be applied to answer questions on problems not explicitly presented in the tutor itself. These findings supported the value of the tutor to produce meaningful learning (Mayer, 2002) or far transfer of learning (Barnett & Ceci, 2002), indicating that informed students could apply general rules to solve novel problems. The research methodology is similar to design-based research (Brown, 1992; Design-Based Research Collective, 2003; Hoadley, 2004) in that instructional design effectiveness was assessed within the context of the classroom over several successive semesters. In assessing meaningful learning over the semesters, the number of rule-based questions was increased, and the