Applications of Programmed Instruction and Interteaching to Technology Education

Henry H. Emurian

Information Systems Department

College of Engineering and Information Technology

UMBC

Baltimore, Maryland 21250

Applications of Programmed Instruction and Interteaching to Technology Education

Miji Mathews, Jingli Wang, Amy Hu, Valeri Scott, John Goodall, Xin Li, Diana Wang, & Lidan Ha
UMBC
&
Ashley G. Durham
Centers for Medicare and Medicaid Services

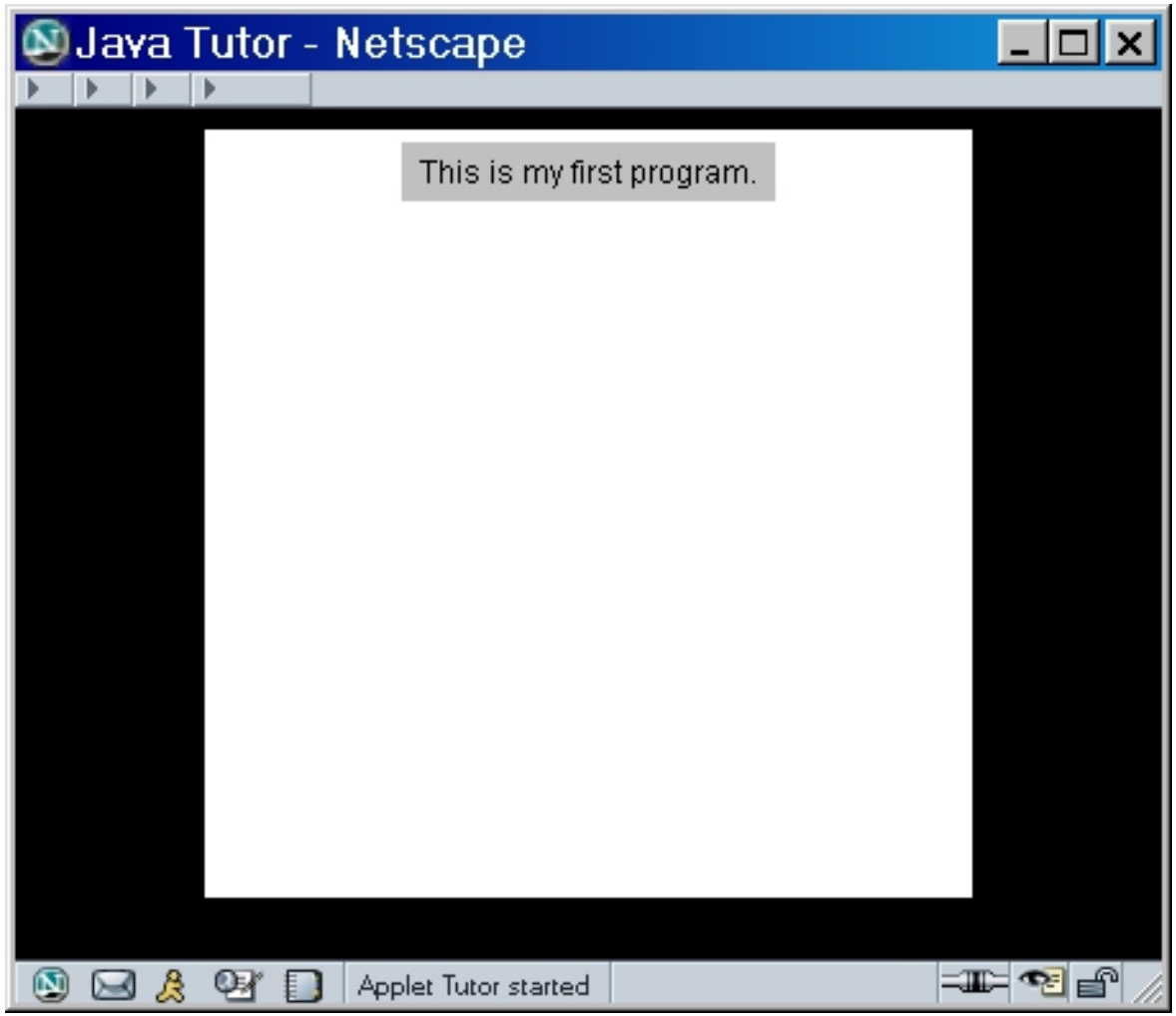
&

Henry H. Emurian
Information Systems Department
College of Engineering and Information Technology
UMBC
Baltimore, Maryland 21250

Performance

```
1.  import java.applet.Applet;
2.  import java.awt.Label;
3.  public class MyProgram extends Applet {
4.  Label myLabel;
5.  public void init() {
6.  myLabel=new Label("This is my first program.");
7.  add(myLabel);
8.  myLabel.setVisible(true);
9.  }
10. }
```

Consequence



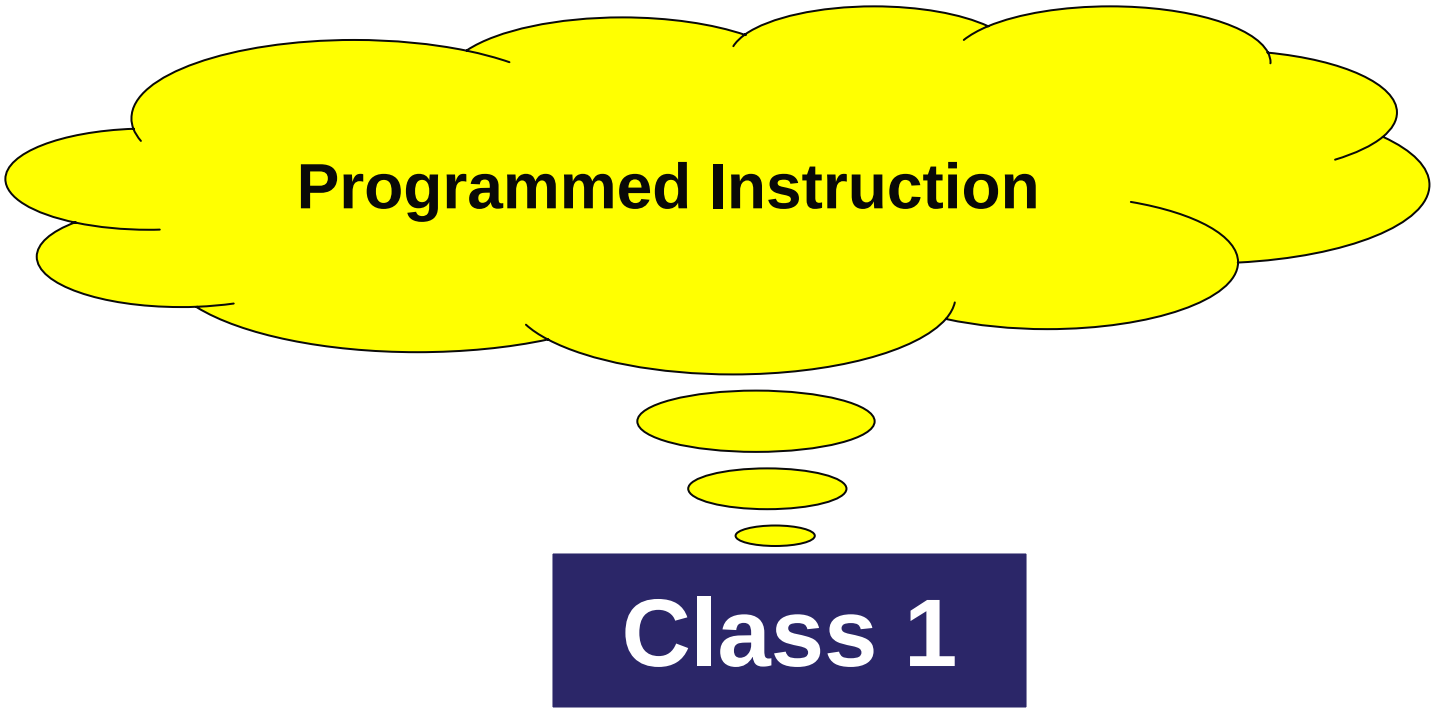
Setting the Stage

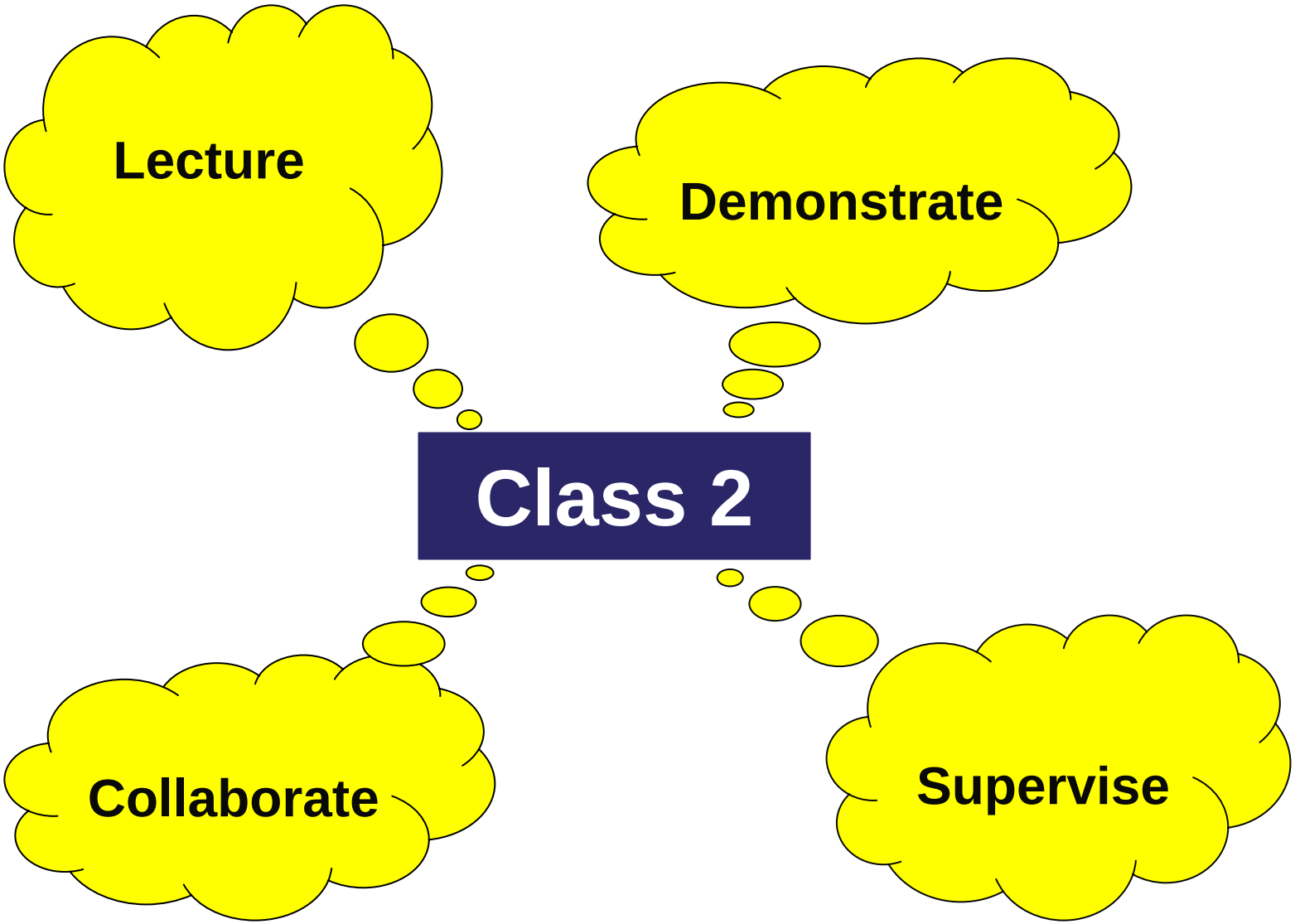
Programmed Instruction

1. A set of **structured interactions** between a learner and a tutor.
2. Occasions **disciplined study behavior** that is focused on the individual learner.
3. Manages the **moment-by-moment interactions** between a learner and a tutor.
4. A **step-wise progression** from elementary knowledge units (*learn units*) or facts to the achievement of a complex repertoire (*meaningful learning*).

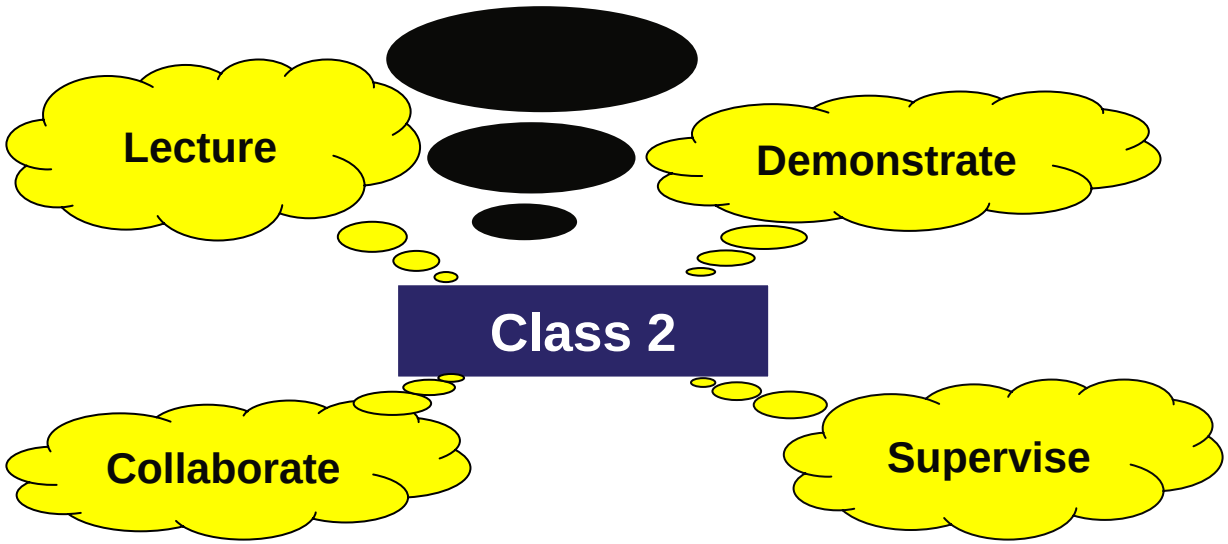
Interteaching

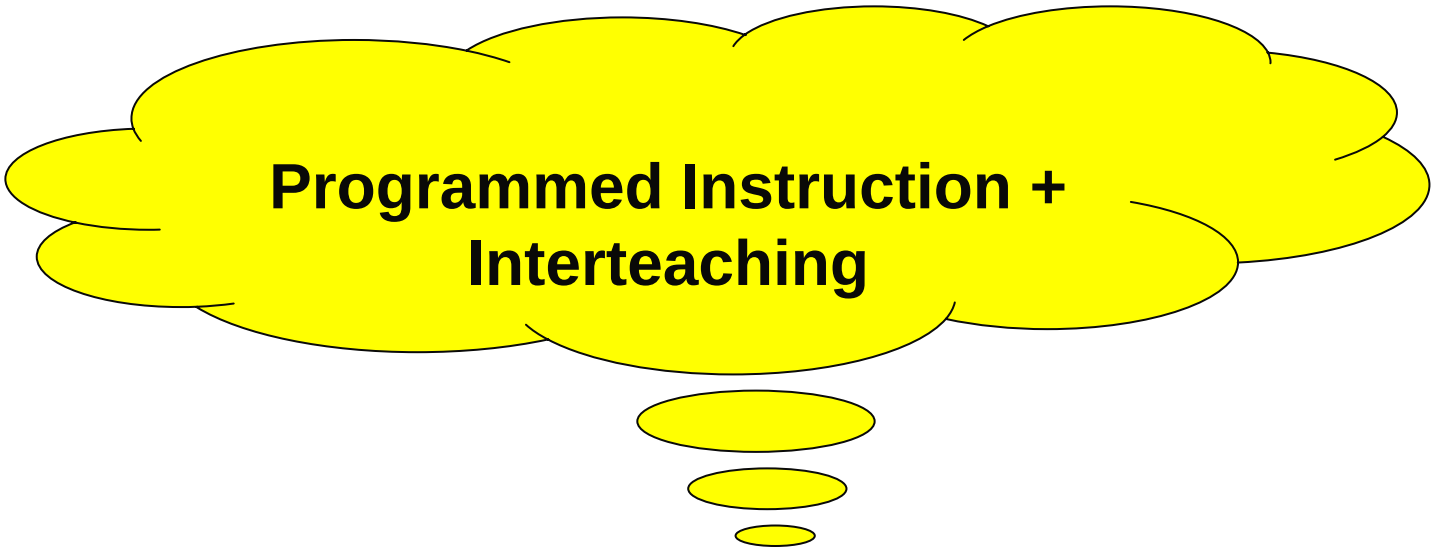
1. A **mutually probing, mutually informing conversation between two people** (Boyce & Hineline, 2002, p. 220).
2. The questions on a topic to be addressed by the participants during a dialogue are prepared in advance by the teacher, and the **students come prepared to interteach.**
3. Has the objective of insuring, by the participants as a team, that **each member of the dyad** can answer the questions with understanding.


Programmed Instruction
Class 1


Lecture
Demonstrate
Class 2
Collaborate
Supervise

Far Transfer Miseries





Class 1
2004

Why is this work hard?

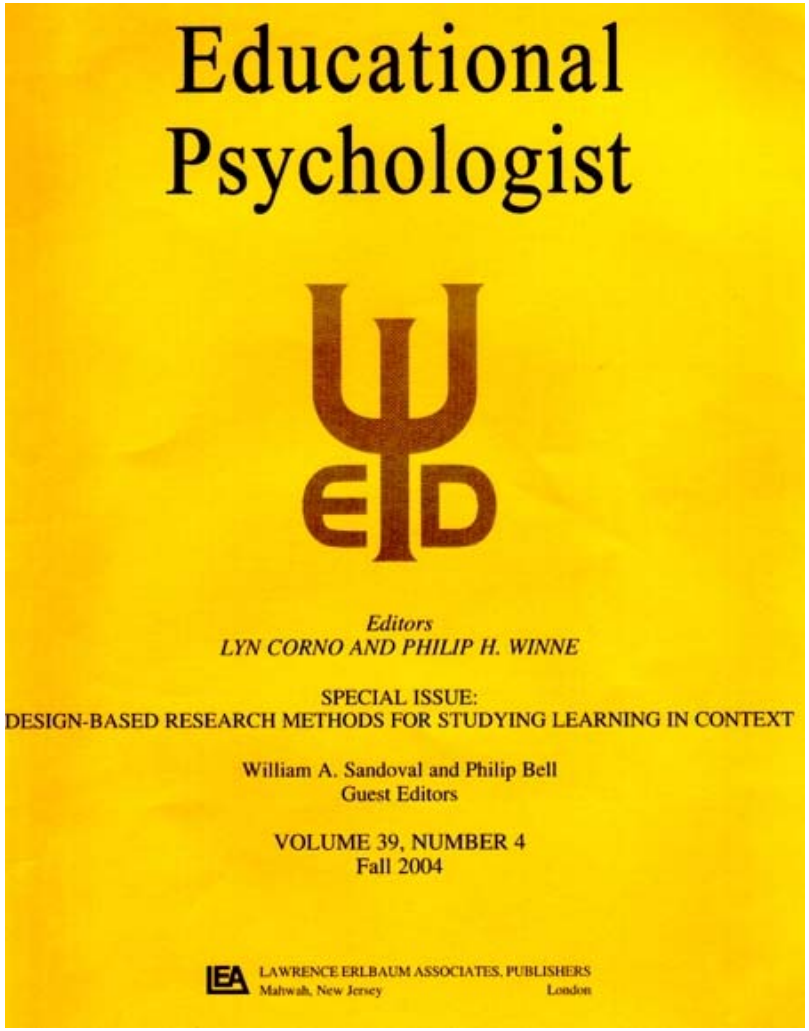
- Educational research is low prestige research.
  - *They Should Know*.
  - Autonomous “man” is alive and well.
- The randomized field trial is now the gold standard.
  - Statistical control is alive and well.
- Overcoming, rather than documenting, individual differences is still viewed with suspicion.
  - Introductory courses in science, technology, engineering, and mathematics (STEM) are for screening.

What do we need?

What We Need

- Big reinforcers for organizational change!

Behavior analysis, by any other name…



Although planned comparisons do occur, the design-based researcher frequently follows new revelations where they lead, tweaking both the intervention and the measurement as the research progresses (Hoadley, 2004, p. 204).

Overview

- Objectives of training
- Challenges in teaching computer programming to IS majors
- A programmed instruction tutoring system
  - From rote memorization to *meaningful learning*
- Interteaching
  - *Let's talk about it*.
- Evidence of effectiveness
- Evidence of students' acceptance and appreciation of this teaching technology

Original Objective, Repeated

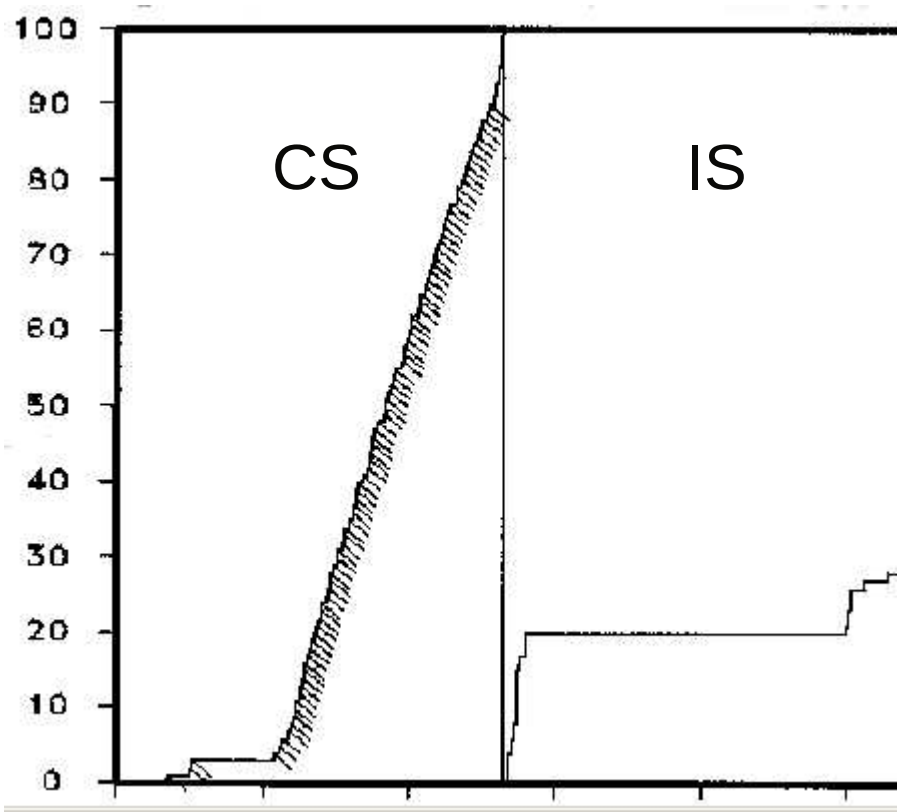
```
1.  import java.applet.Applet;
2.  import java.awt.Label;
3.  public class MyProgram extends Applet {
4.  Label myLabel;
5.  public void init() {
6.  myLabel=new Label("This is my first program.");
7.  add(myLabel);
8.  myLabel.setVisible(true);
9.  }
10. }
```

Original Solution (circa 2000)

- ***Lecture***
  - Write the code on the board and explain it.
- ***Exhort***
  - Tell the students to learn a program for a test.
- ***Test***
- ***Observe***
  - Individual differences in test performance.
- ***Repeat the Above***

Why won't they respond?

- In comparison to Computer Science (CS) students, Information Systems (IS) students exhibit a low rate of computer programming.



Challenges

- Students in Information Systems (IS) do **not** like to write computer programs.
- IS students have **minimal coursework** in computer programming and programming languages.
- IS students **need** a fundamental mastery of programming principles, especially related to the object-oriented paradigm.
- IS students are often **demoralized** by taking courses with computer science majors taught by computer science faculty.
- How can we **help** IS students achieve the objective?

Emergent Insight

- **IS students are *insensitive to reinforcers* that are symbols.**
- **IS students lack a history that produces *conditioned reinforcers* that can sustain learning.**

Semi-Enlightened Solution

Objective

```
1.  import java.applet.Applet;
2.  import java.awt.Label;
3.  public class MyProgram
    extends Applet{
4.  Label myLabel;
5.  public void init(){
6.  myLabel=new Label("This is
    my first program.");
7.  add(myLabel);
8.  myLabel.setVisible(true);
9.  }
10. }
```

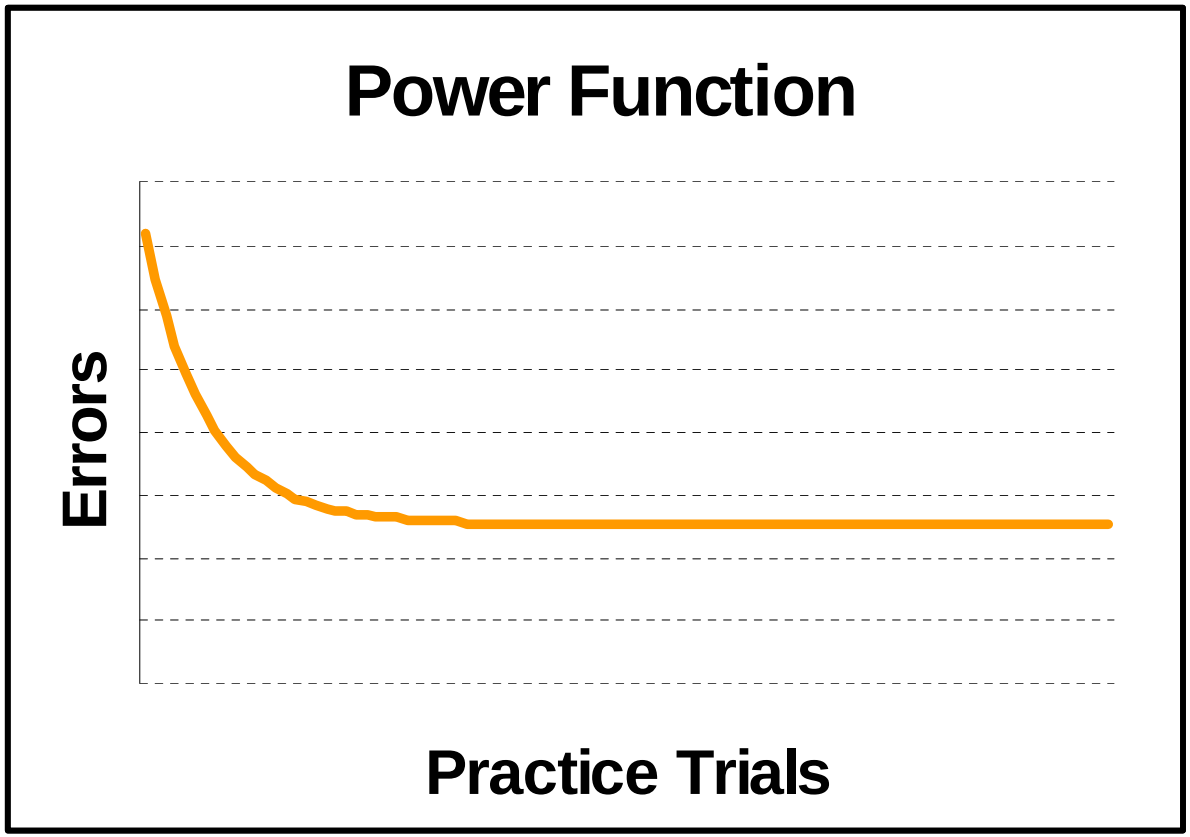
Solution (circa 2000)

- **Programmed instruction**
- **Modified personalized system of instruction**

Underlying Assistive Principles

1. **Rote Memorization is Good**
   - Fundamental to meaningful learning
     - Near and far transfer
   - Constructivism comes later (much, much later...).
2. **Disciplined Study Behavior is Good**
   - It is essential to mastery.
   - Most students don't know how to study. They don't know how to monitor their acquisition of competence.
3. **Repetition and Overlearning are Good**
   - Contribute to retention
4. **A Steady State is Good**
5. **Feeling Good is Good**
   - Sustains hard work and encourages future learning.

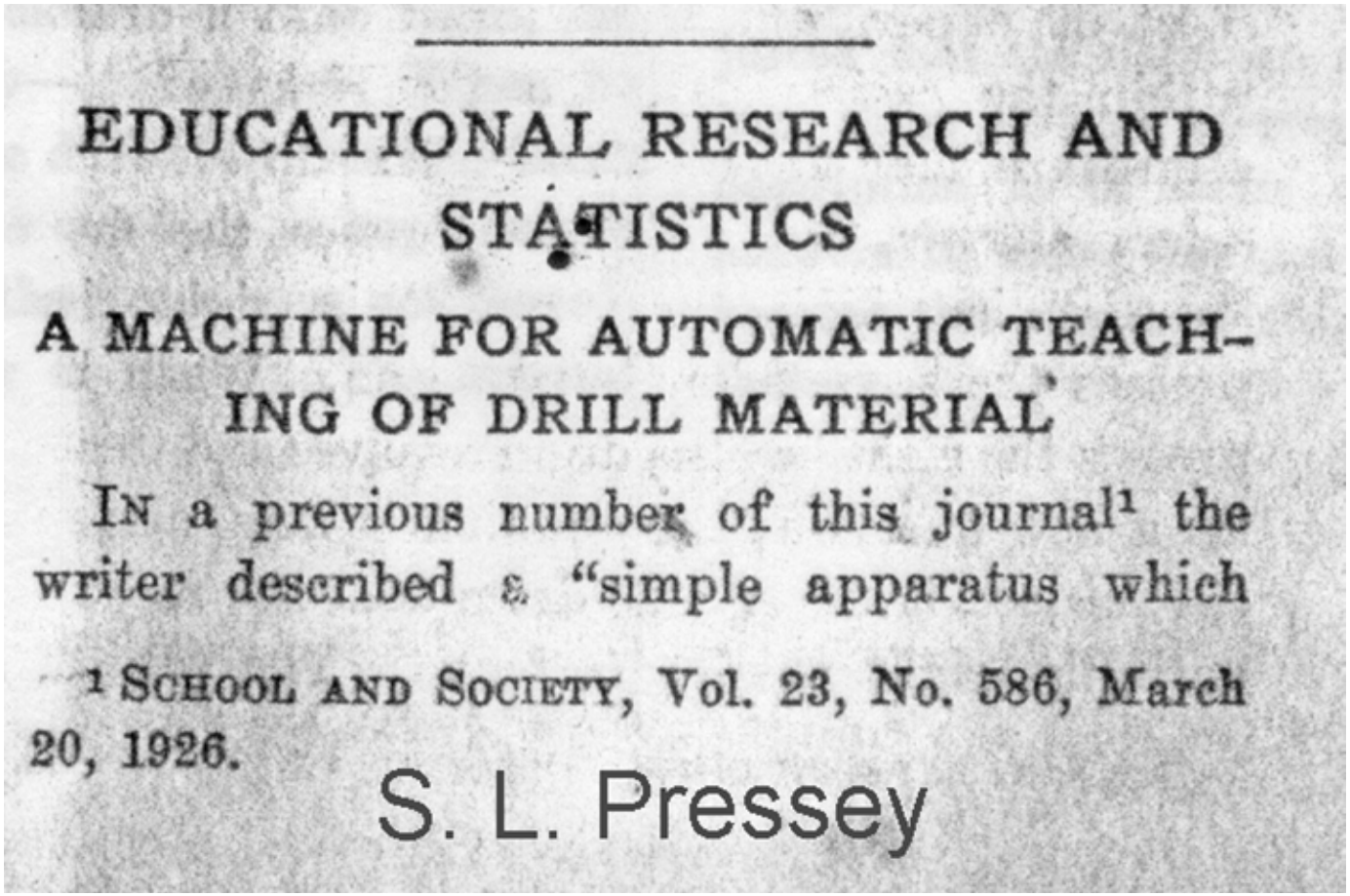
What state is steady?



Enlightened Objectives

- Meaningful learning
  - Transfer Near …and Far
    - Equivalence classes
      - Beyond generic extensions of the tact
- Rule-Governed behavior
  - Relationally framing
    - Response generality through hierarchical combinatorial entailment
    - Emitting behavior that is correct but that has never entered into a 3-term contingency relationship.

Pressey



EDUCATIONAL RESEARCH AND STATISTICS

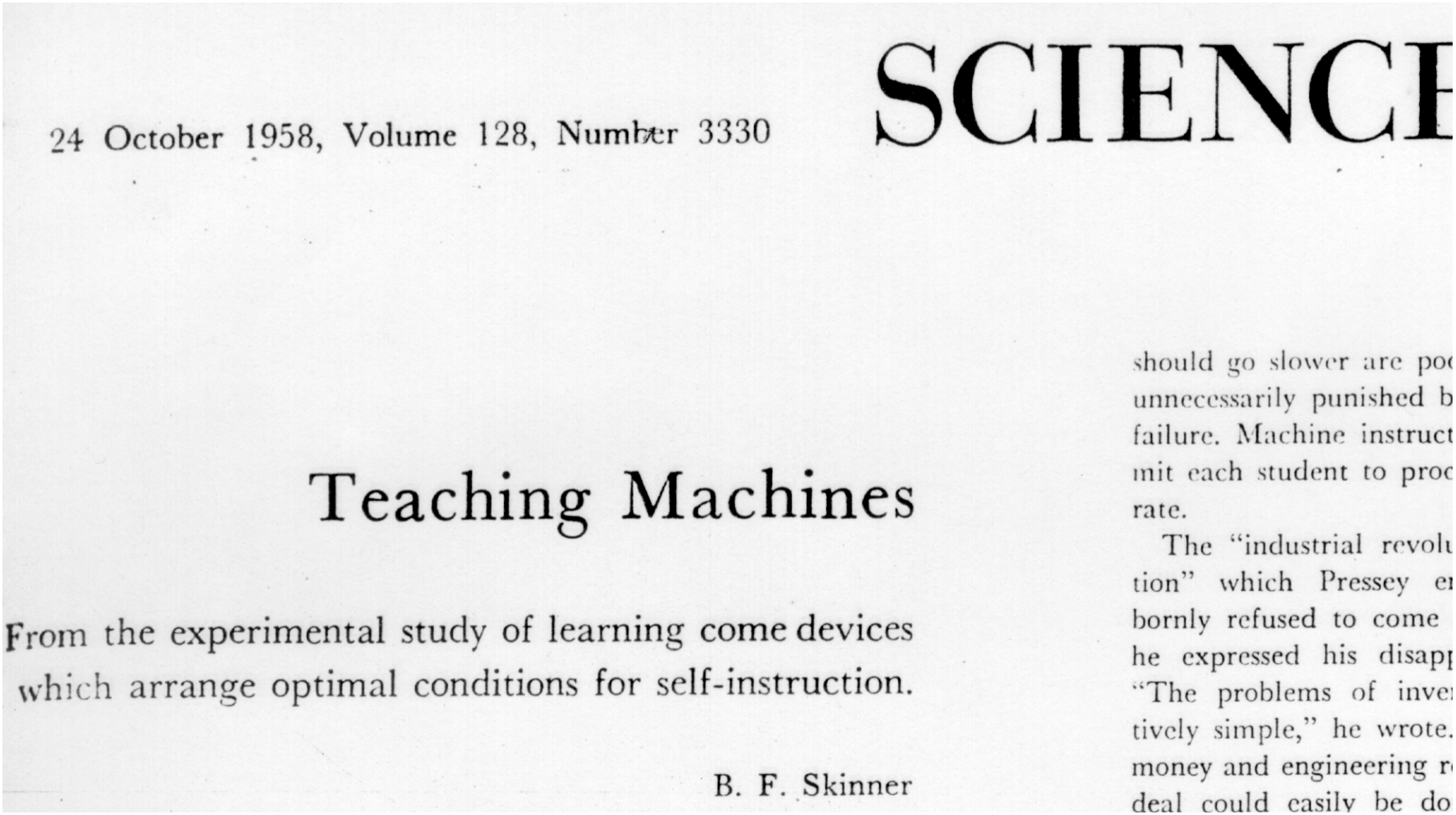
A MACHINE FOR AUTOMATIC TEACHING OF DRILL MATERIAL

IN a previous number of this journal[1] the writer described a "simple apparatus which

[1] SCHOOL AND SOCIETY, Vol. 23, No. 586, March 20, 1926.

S. L. Pressey

Skinner



24 October 1958, Volume 128, Number 3330

SCIENCE

Teaching Machines

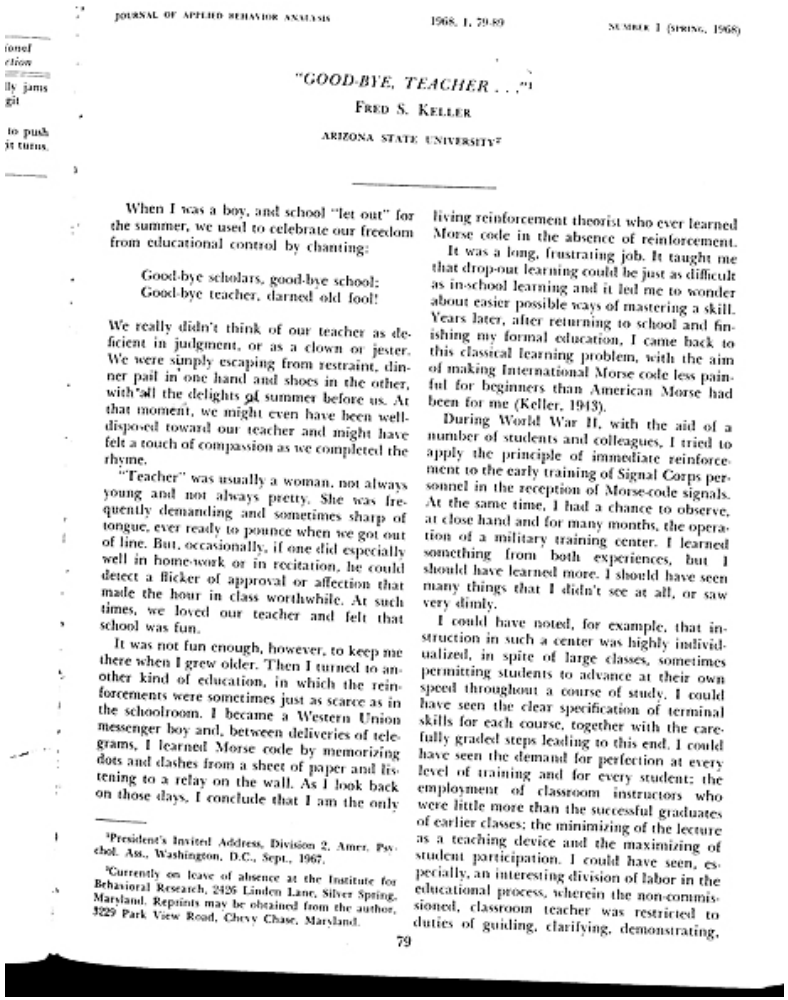
From the experimental study of learning come devices which arrange optimal conditions for self-instruction.

B. F. Skinner

should go slower are poo
unnecessarily punished b
failure. Machine instruct
mit each student to proc
rate.

The "industrial revolu
tion" which Pressey e
bornly refused to come
he expressed his disapp
"The problems of inve
tively simple," he wrote.
money and engineering r
deal could easily be do

Keller



JOURNAL OF APPLIED BEHAVIOR ANALYSIS 1968, 1, 79-89 NUMBER 1 (SPRING, 1968)

"GOOD-BYE, TEACHER . . ."[1]

FRED S. KELLER

ARIZONA STATE UNIVERSITY[2]

When I was a boy, and school "let out" for the summer, we used to celebrate our freedom from educational control by chanting:

Good-bye scholars, good-bye school;
Good-bye teacher, darned old fool!

We really didn't think of our teacher as deficient in judgment, or as a clown or jester. We were simply escaping from restraint, dinner pail in one hand and shoes in the other, with all the delights of summer before us. At that moment, we might even have been well-disposed toward our teacher and might have felt a touch of compassion as we completed the rhyme.

"Teacher" was usually a woman, not always young and not always pretty. She was frequently demanding and sometimes sharp of tongue, ever ready to pounce when we got out of line. But, occasionally, if one did especially well in home-work or in recitation, he could detect a flicker of approval or affection that made the hour in class worthwhile. At such times, we loved our teacher and felt that school was fun.

It was not fun enough, however, to keep me there when I grew older. Then I turned to another kind of education, in which the reinforcements were sometimes just as scarce as in the schoolroom. I became a Western Union messenger boy and, between deliveries of telegrams, I learned Morse code by memorizing dots and dashes from a sheet of paper and listening to a relay on the wall. As I look back on those days, I conclude that I am the only living reinforcement theorist who ever learned Morse code in the absence of reinforcement.

It was a long, frustrating job. It taught me that drop-out learning could be just as difficult as in-school learning and it led me to wonder about easier possible ways of mastering a skill. Years later, after returning to school and finishing my formal education, I came back to this classical learning problem, with the aim of making International Morse code less painful for beginners than American Morse had been for me (Keller, 1943).

During World War II, with the aid of a number of students and colleagues, I tried to apply the principle of immediate reinforcement to the early training of Signal Corps personnel in the reception of Morse-code signals. At the same time, I had a chance to observe, at close hand and for many months, the operation of a military training center. I learned something from both experiences, but I should have learned more. I should have seen many things that I didn't see at all, or saw very dimly.

I could have noted, for example, that instruction in such a center was highly individualized, in spite of large classes, sometimes permitting students to advance at their own speed throughout a course of study. I could have seen the clear specification of terminal skills for each course, together with the carefully graded steps leading to this end. I could have seen the demand for perfection at every level of training and for every student: the employment of classroom instructors who were little more than the successful graduates of earlier classes; the minimizing of the lecture as a teaching device and the maximizing of student participation. I could have seen, especially, an interesting division of labor in the educational process, wherein the non-commissioned, classroom teacher was restricted to duties of guiding, clarifying, demonstrating,

[1]President's Invited Address, Division 2, Amer. Psychol. Ass., Washington, D.C., Sept., 1967.

[2]Currently on leave of absence at the Institute for Behavioral Research, 2426 Linden Lane, Silver Spring, Maryland. Reprints may be obtained from the author, 3229 Park View Road, Chevy Chase, Maryland.

79

Greer

The Behavior Analyst 1999, 22, 5–16 No. 1 (Spring)

Is the Learn Unit a Fundamental Measure of Pedagogy?

R. Douglas Greer
Columbia University Teachers College

Sally Hogin McDonough
The Fred S. Keller School

We propose a measure of teaching, the *learn unit*, that explicitly describes the interaction between teachers and their students. The theoretical, educational research, and applied behavior analysis literatures all converge on the learn unit as a fundamental measure of teaching. The theoretical literature proposes the construct of the *interlocking operant* and embraces verbal behavior, social interaction, and translations of psychological constructs into complex theoretical respondent-operant interactions and behavior–behavior relations. Research findings in education and applied behavior analysis on engaged academic time, opportunity to respond, active student responding, teacher–student responding, student–teacher responding, tutor–tutee responding, tutee–tutor responding, and verbal episodes between individuals all support a measure of interlocking responses. More recently, research analyzing the components of both the students' and teachers' behavior suggests that the learn unit is the strongest predictor of effective teaching. Finally, we propose applications of the learn unit to other issues in pedagogy not yet researched and the relation of learn units to the verbal behavior of students.

Key words: operant, three-term contingency, opportunity to respond, pedagogical measurement unit, interlocking operants, cost-benefit analyses

Dyadic Collaboration

Computer Science Education
2002, Vol. 12, No. 3, pp. 197–212

0899-3408/02/1203-197$16.00

In Support of Pair Programming in the Introductory Computer Science Course

Laurie Williams[1], Eric Wiebe[2], Kai Yang[1], Miriam Ferzli[2], and Carol Miller[1]
North Carolina State University, Raleigh, NC, USA, [1]Department of Computer Science, and [2]Department of Math, Science and Technology Education

ABSTRACT

A formal pair programming experiment was run at North Carolina to empirically assess the educational efficacy of the technique in a CS1 course. Results indicate that students who practice pair programming perform better on programming projects and are more likely to succeed by completing the class with a C or better. Student pairs are more self-sufficient which reduces their reliance on the teaching staff. Qualitatively, paired students demonstrate higher order thinking skills than students who work alone. These results are supportive of pair programming as a collaborative learning technique.

Interteaching

The Behavior Analyst 2002, 25, 215–226 No. 2 (Fall)

Interteaching: A Strategy for Enhancing the User-Friendliness of Behavioral Arrangements in the College Classroom

Thomas E. Boyce
University of Nevada, Reno

Philip N. Hineline
Temple University

"Interteaching" is an arrangement for college classroom instruction that departs from the standard lecture format and offers an answer to criticisms commonly directed at behavioral teaching techniques. This approach evolved from exploratory use of small-group arrangements and Ferster and Perrott's (1968) "interview technique," leading ultimately to a format that is organized around focused dyadic discussion. Specific suggestions are offered that might enable both seasoned and novice instructors to incorporate this or similar arrangements into their classrooms. This approach retains some key characteristics of Keller's personalized system of instruction and precision teaching, but offers greater flexibility for strategies that are based on behavioral principles.

Key words: applied behavior analysis, education, instruction, interviewing, PSI, precision teaching, reciprocal peer tutoring

Me

PERGAMON

Computers in Human Behavior (2003)

Computers in Human Behavior

www.elsevier.com/locate/comphumbeh

A programmed instruction tutoring system for Java™: consideration of learning performance and software self-efficacy

Henry H. Emurian*

Information Systems Department, UMBC, 1000 Hilltop Circle, Baltimore, MD 21250, USA

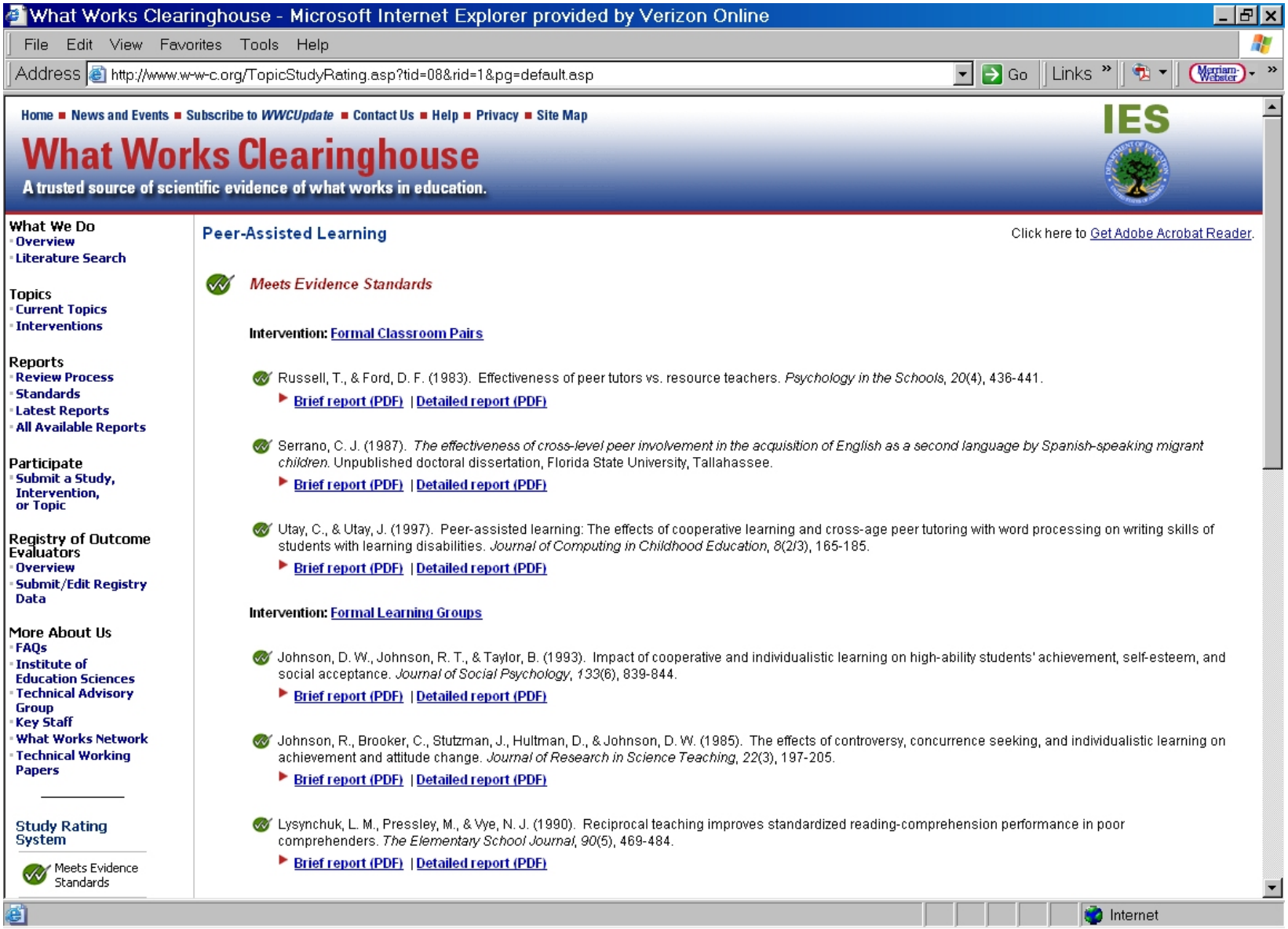
Abstract

An undergraduate ($n = 23$) and a graduate ($n = 23$) class of information systems majors used a Web-based tutoring system during the first 3-h session of a 14-week course in interface design and implementation. The tutoring system taught a simple Java™ applet as the first technical training exercise, and the instructional design was based upon programmed instruction, which is a competency-based tutoring system. Software self-efficacy was assessed prior to using the tutor and at the end of the 3-h period. Students' interactive performances (errors and help selections) were recorded for all interfaces in the tutor. The results showed that the undergraduate students made more input and test errors than did the graduate students, but the number of students in each class who completed all eight tutor stages (18 undergraduates and 17 graduates) was almost equivalent. Forty-four of the 46 students completed the fourth tutor stage, which presented frames of information explaining the items in the program. Students who did not complete all eight stages showed more errors on the initial four stages, in comparison to students who did complete all stages. Software self-efficacy increased from pre-tutor to post-tutor occasions for both classes and for both completers and non-completers. No significant relationship was found between software self-efficacy changes and tutor learning performance. Neither was gender related to software self-efficacy changes or learning performance. Evaluations of the tutor were favorable by almost all learners. A competency-based tutoring system may produce both skill and earned self-efficacy at the level of the individual learner, without regard to variations in the learning process leading to mastery.

Keywords: Computer training; Programmed instruction; Software self-efficacy

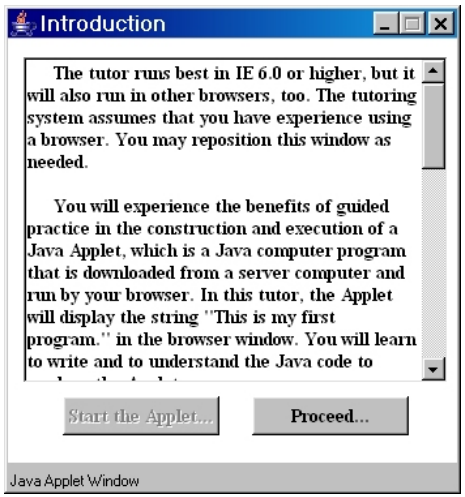
* Tel.: +1-410-455-3206; fax: +1-410-455-1073.
E-mail address: emurian@umbc.edu (H.H. Emurian).

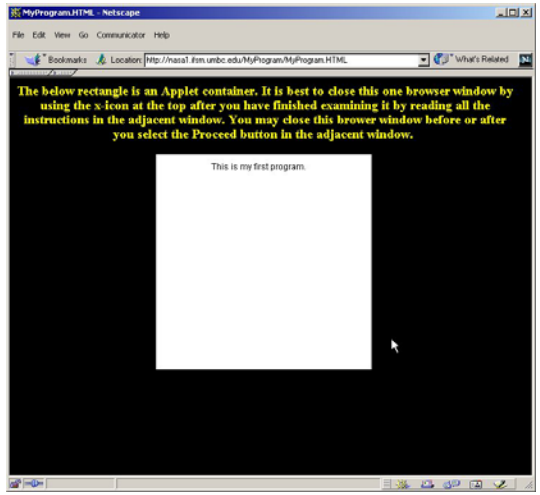
0747-5632/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0747-5632(03)00048-7


What Works Clearinghouse - Microsoft Internet Explorer provided by Verizon Online
File Edit View Favorites Tools Help
Address http://www.w-w-c.org/TopicStudyRating.asp?tid=08&rid=1&pg=default.asp
Go Links
Home ■ News and Events ■ Subscribe to WWCUpdate ■ Contact Us ■ Help ■ Privacy ■ Site Map
What Works Clearinghouse
A trusted source of scientific evidence of what works in education.
IES
What We Do
Overview
Literature Search
Topics
Current Topics
Interventions
Reports
Review Process
Standards
Latest Reports
All Available Reports
Participate
Submit a Study, Intervention, or Topic
Registry of Outcome Evaluators
Overview
Submit/Edit Registry Data
More About Us
FAQs
Institute of Education Sciences
Technical Advisory Group
Key Staff
What Works Network
Technical Working Papers
Study Rating System
Meets Evidence Standards
Peer-Assisted Learning
Click here to Get Adobe Acrobat Reader.
Meets Evidence Standards
Intervention: Formal Classroom Pairs
Russell, T., & Ford, D. F. (1983). Effectiveness of peer tutors vs. resource teachers. Psychology in the Schools, 20(4), 436-441.
Brief report (PDF) | Detailed report (PDF)
Serrano, C. J. (1987). The effectiveness of cross-level peer involvement in the acquisition of English as a second language by Spanish-speaking migrant children. Unpublished doctoral dissertation, Florida State University, Tallahassee.
Brief report (PDF) | Detailed report (PDF)
Utay, C., & Utay, J. (1997). Peer-assisted learning: The effects of cooperative learning and cross-age peer tutoring with word processing on writing skills of students with learning disabilities. Journal of Computing in Childhood Education, 8(2/3), 165-185.
Brief report (PDF) | Detailed report (PDF)
Intervention: Formal Learning Groups
Johnson, D. W., Johnson, R. T., & Taylor, B. (1993). Impact of cooperative and individualistic learning on high-ability students' achievement, self-esteem, and social acceptance. Journal of Social Psychology, 133(6), 839-844.
Brief report (PDF) | Detailed report (PDF)
Johnson, R., Brooker, C., Stutzman, J., Hultman, D., & Johnson, D. W. (1985). The effects of controversy, concurrence seeking, and individualistic learning on achievement and attitude change. Journal of Research in Science Teaching, 22(3), 197-205.
Brief report (PDF) | Detailed report (PDF)
Lysynchuk, L. M., Pressley, M., & Vye, N. J. (1990). Reciprocal teaching improves standardized reading-comprehension performance in poor comprehenders. The Elementary School Journal, 90(5), 469-484.
Brief report (PDF) | Detailed report (PDF)
Internet

Programmed Instruction Tutoring System

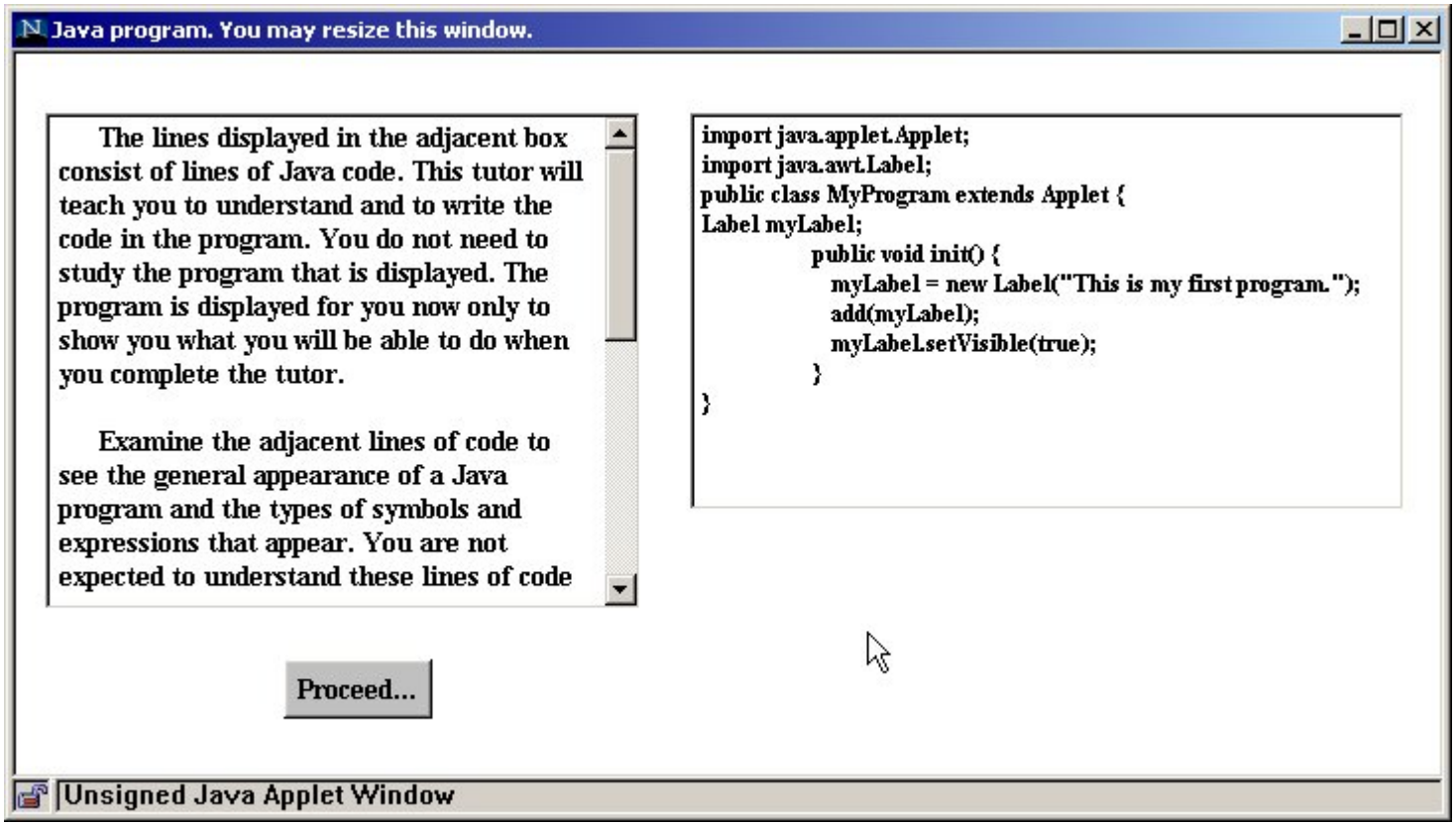
Introduction



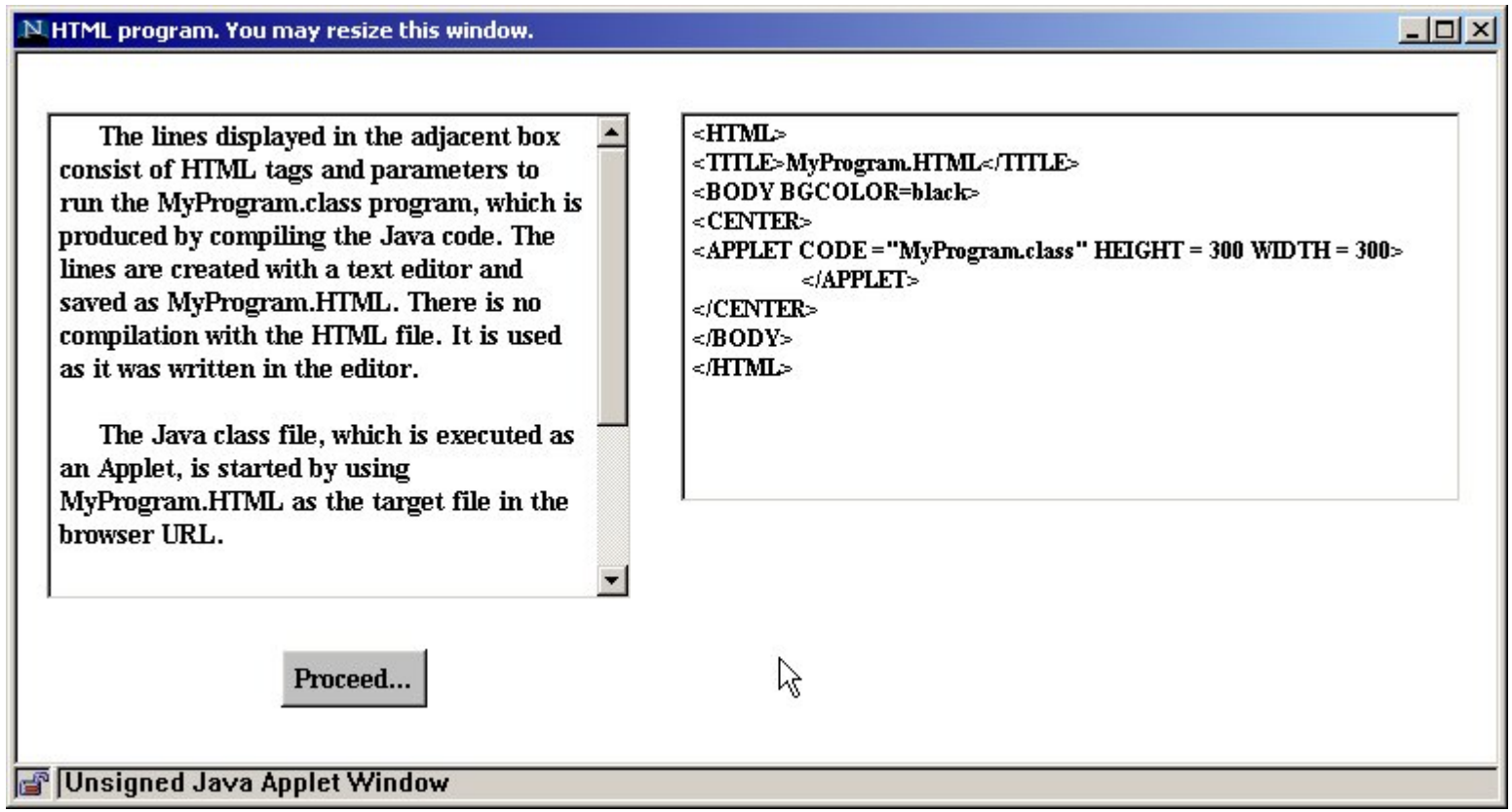


- Advance organizers
  - Template of a Java Applet
- Observe Applet in action

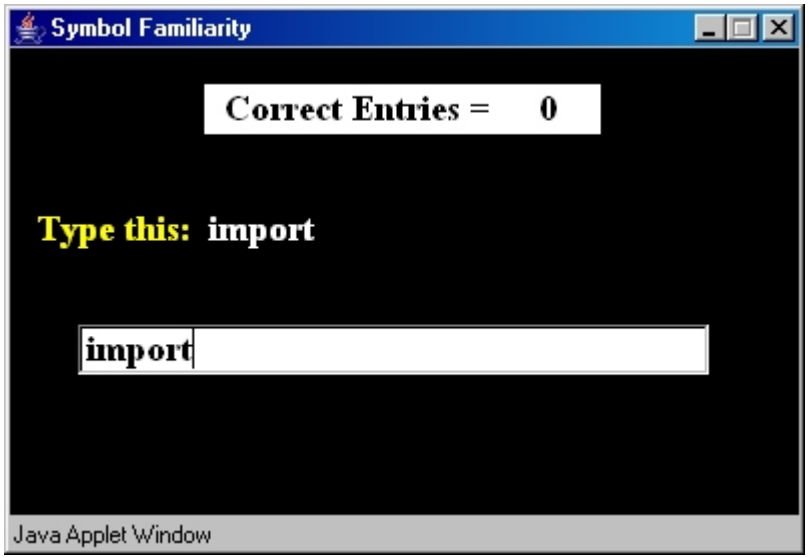
Applet Code Orientation



HTML Orientation

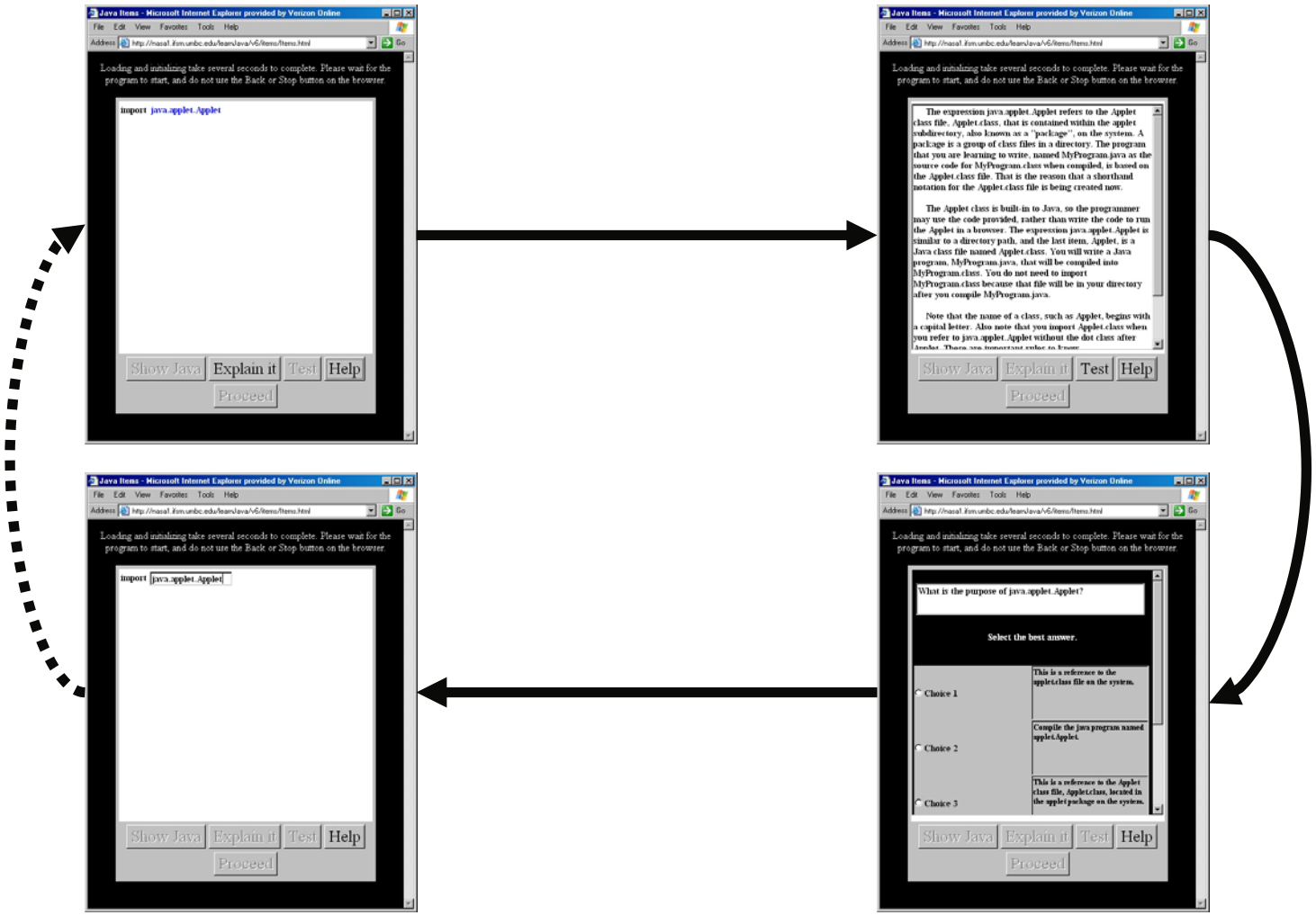


Textual Imitation



Discrimination Training

Item Learning

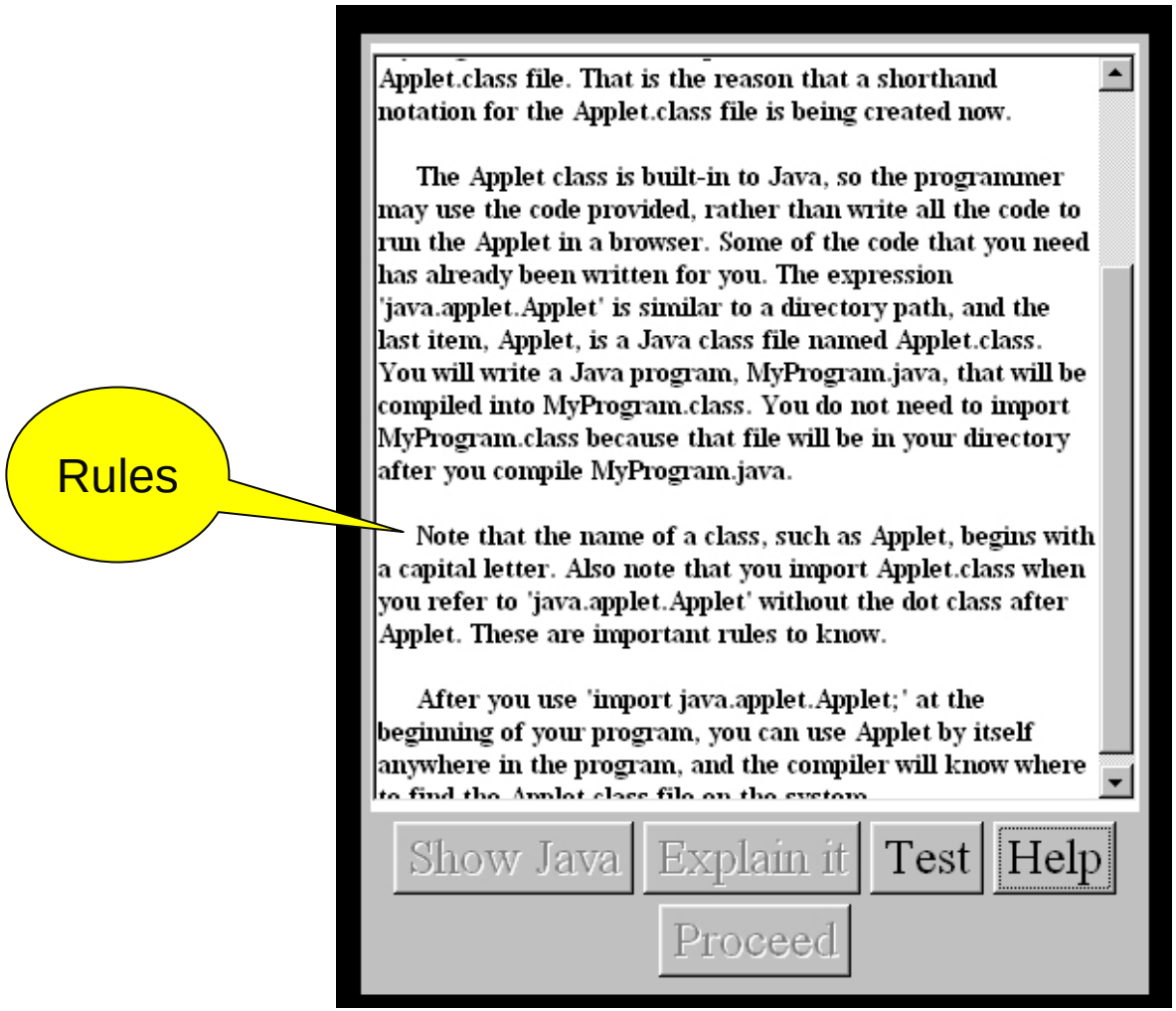


Input Field

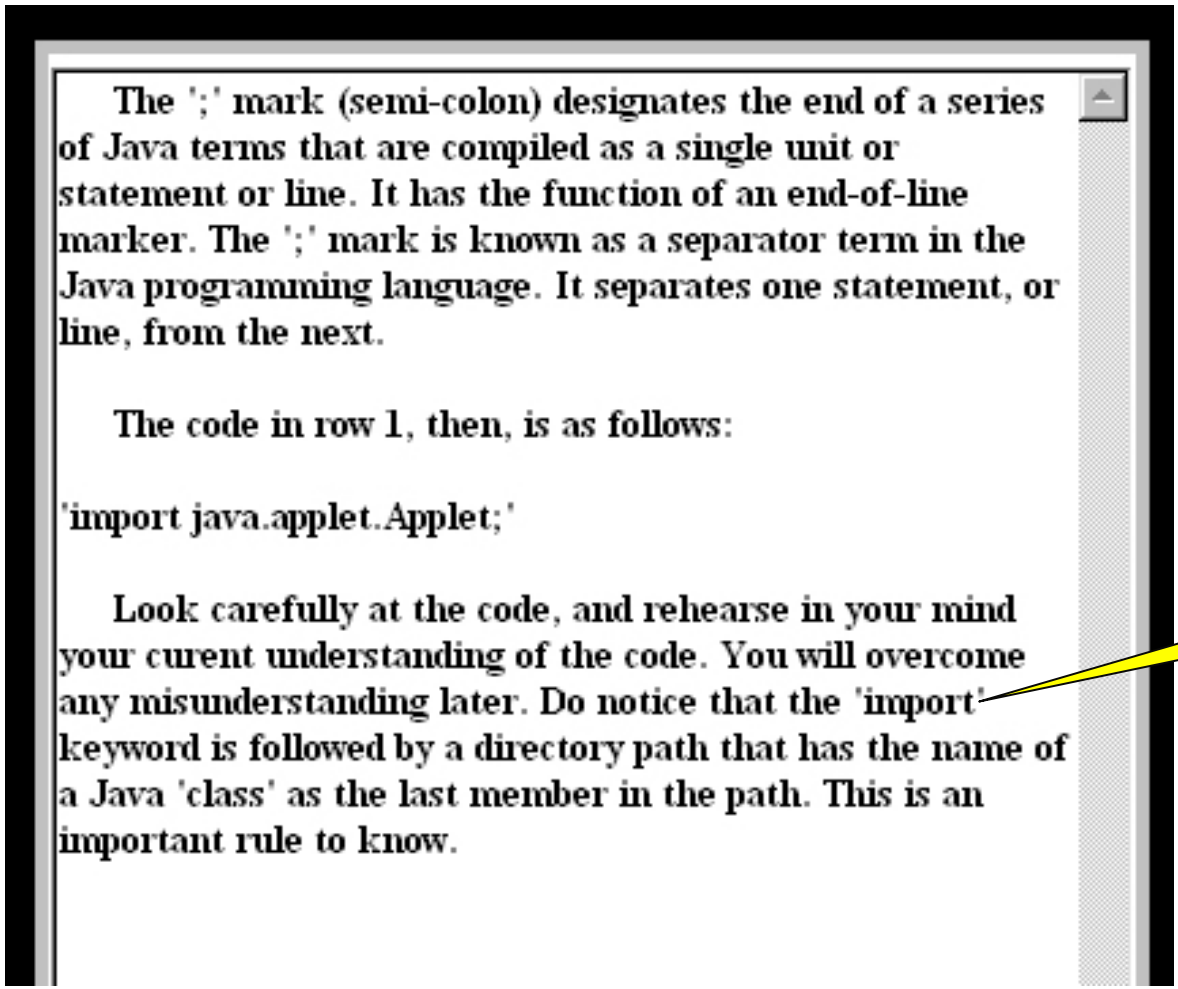
Content Design Considerations

- Fostering Rule-Governed Behavior
  - Embedding “signals” of important rules
  - Embedding reflection prompts
  - Designing tests to promote response generality

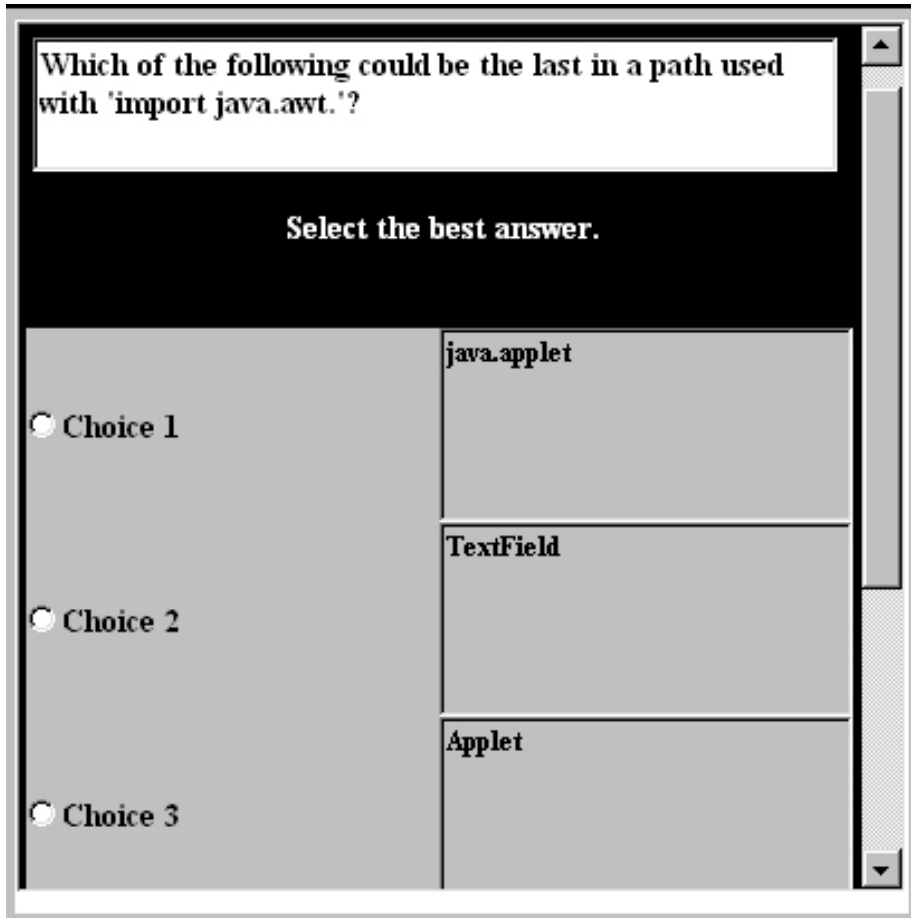
Item Rule Example 1



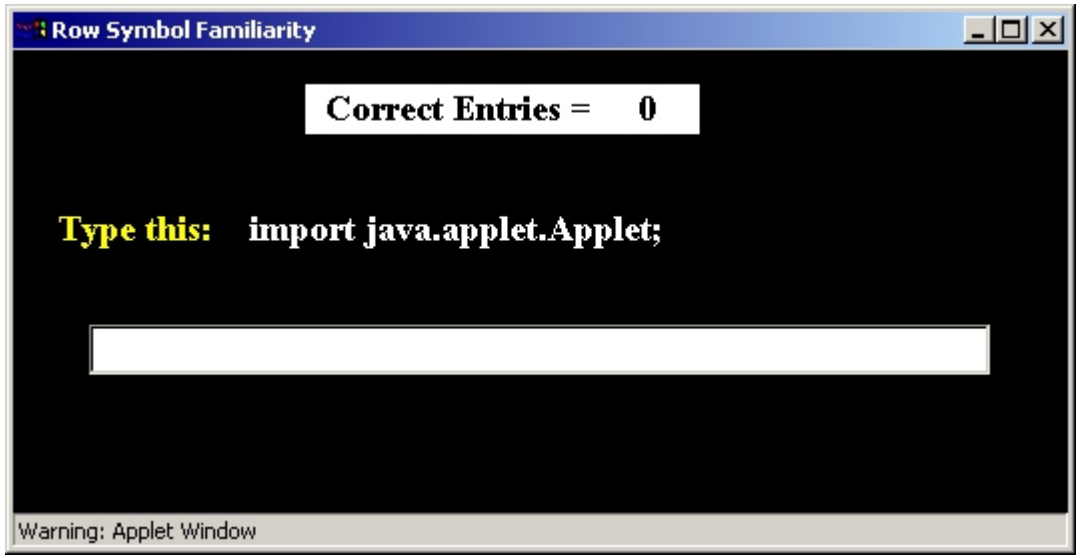
Item Rule Example 2



Test Example

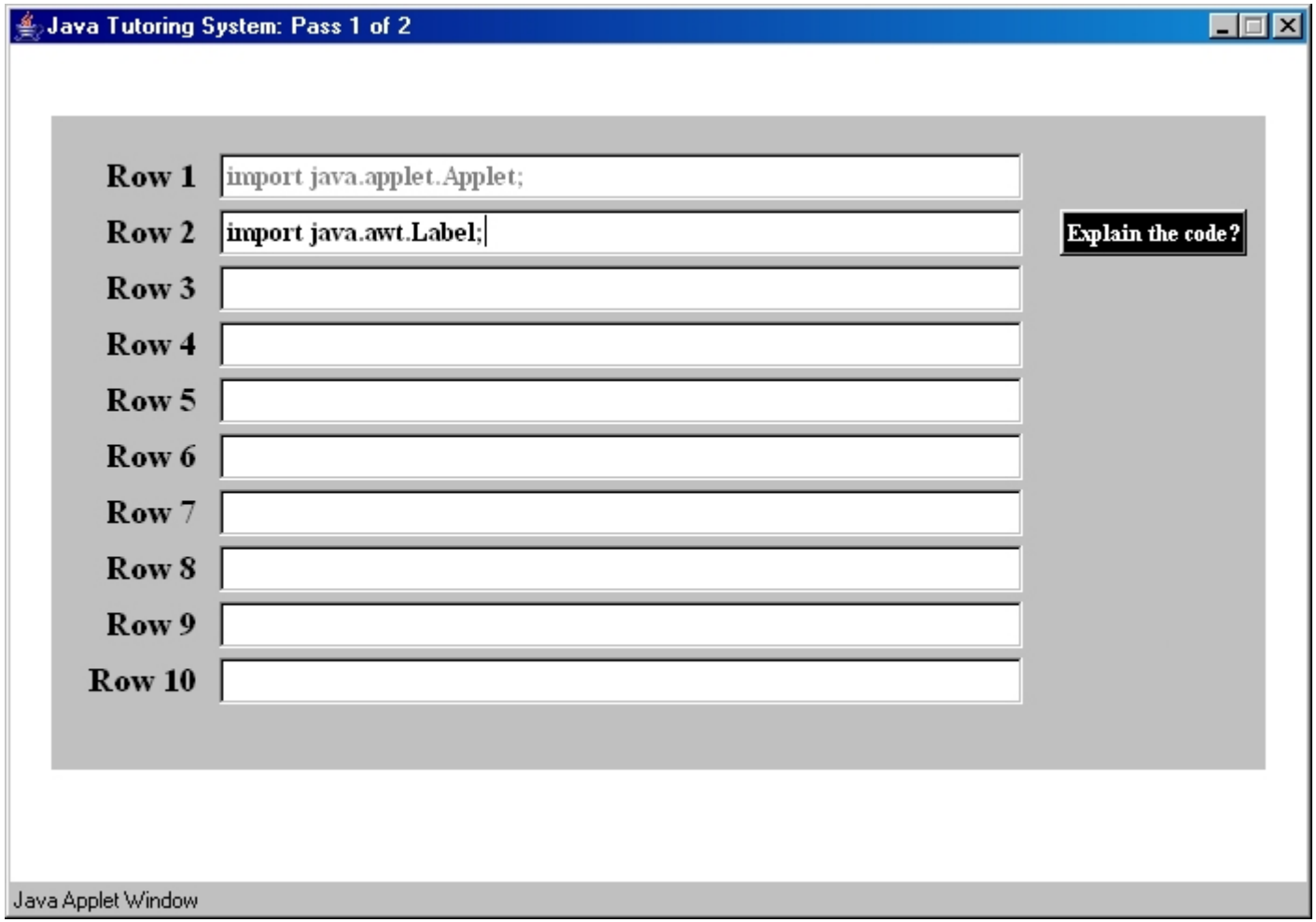


Row Imitation

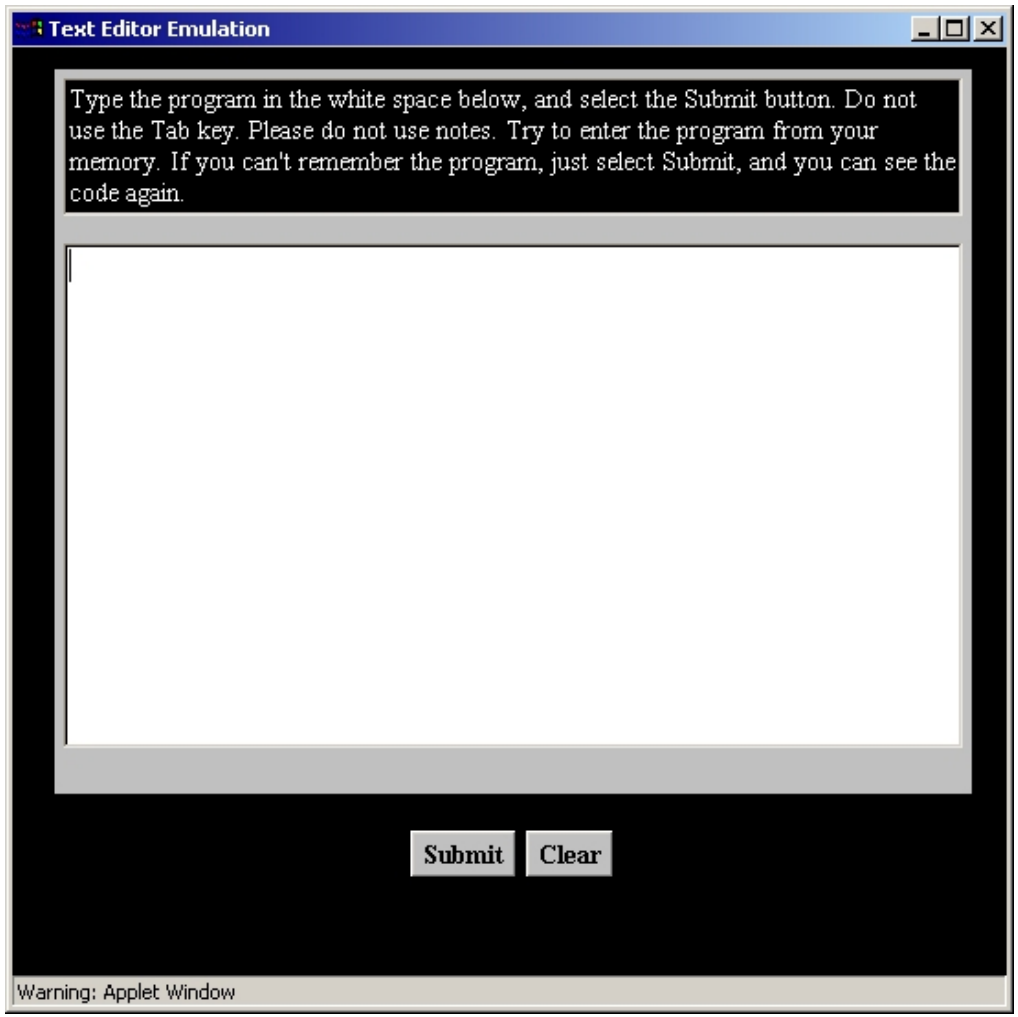


Row Discrimination Training

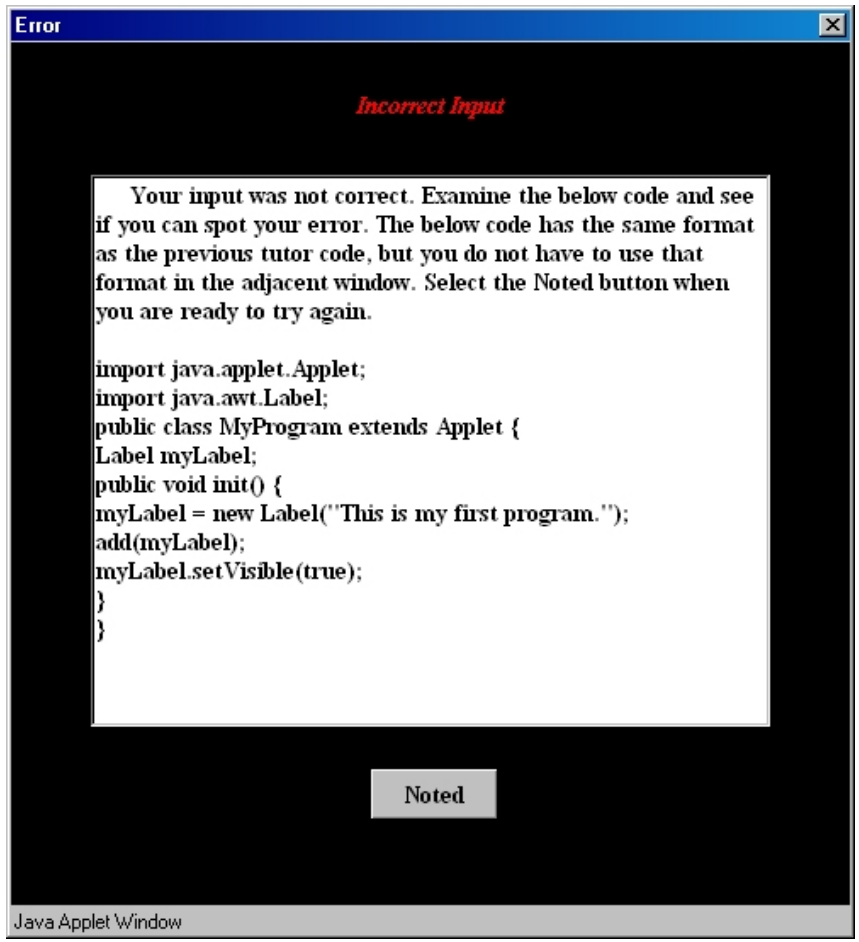
Row Learning



Program Interface



Program Interface



First Interteaching Report

IFSM 413

Interteaching Report #1

Your name xxxx Date 9/8/04

Your partner's name: yyyy

You should understand the components of the below program at a level given in the Java Tutor. Discuss these components with the intention to understand the specific item and any general principle that is reflected in an item or collection of items. An example of a general principle would be to begin the name of a class with a capital letter.

```
import java.applet.Applet;
import java.awt.Label;
public class MyProgram extends Applet {
Label myLabel;
public void init() {
myLabel = new Label("This is my first program.");
add(myLabel);
myLabel.setVisible(true);
}
}
```

How effective was this session in helping you to learn the material?

1 = Not at all effective. The session did not contribute to my learning of the material.

10 = Totally effective. The session contributed to my learning of the material.

(Not effective) 1 2 3 4 5 6 7 8 9 10 (Totally effective)

Enter one number that describes the effectiveness for you: 9.

How confident are you that you could answer all questions correctly if you were tested on this program right now?

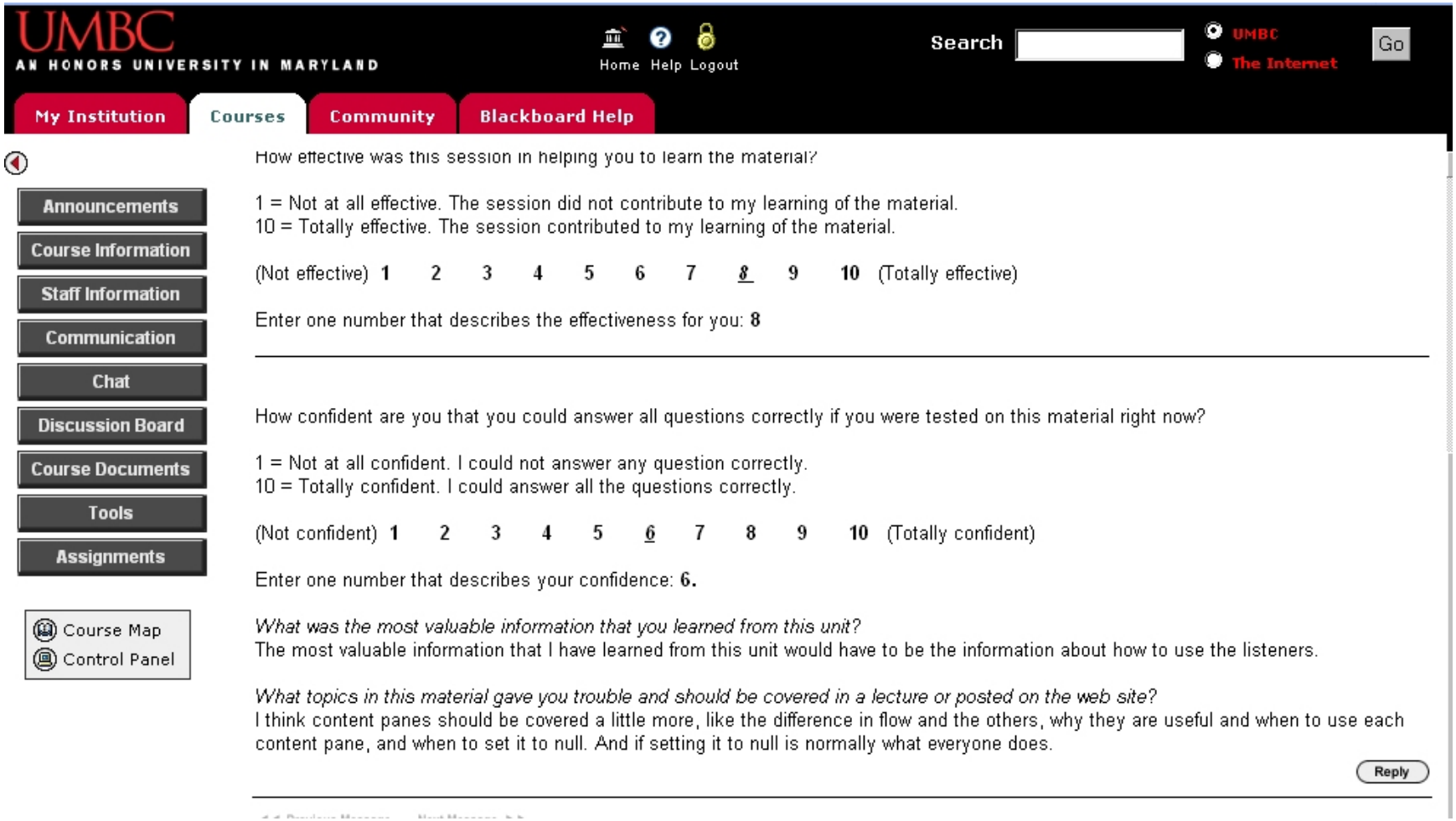
1 = Not at all confident. I could not answer any question correctly.

10 = Totally confident. I could answer all the questions correctly.

(Not confident) 1 2 3 4 5 6 7 8 9 10 (Totally confident)

Enter one number that describes your confidence: 9.

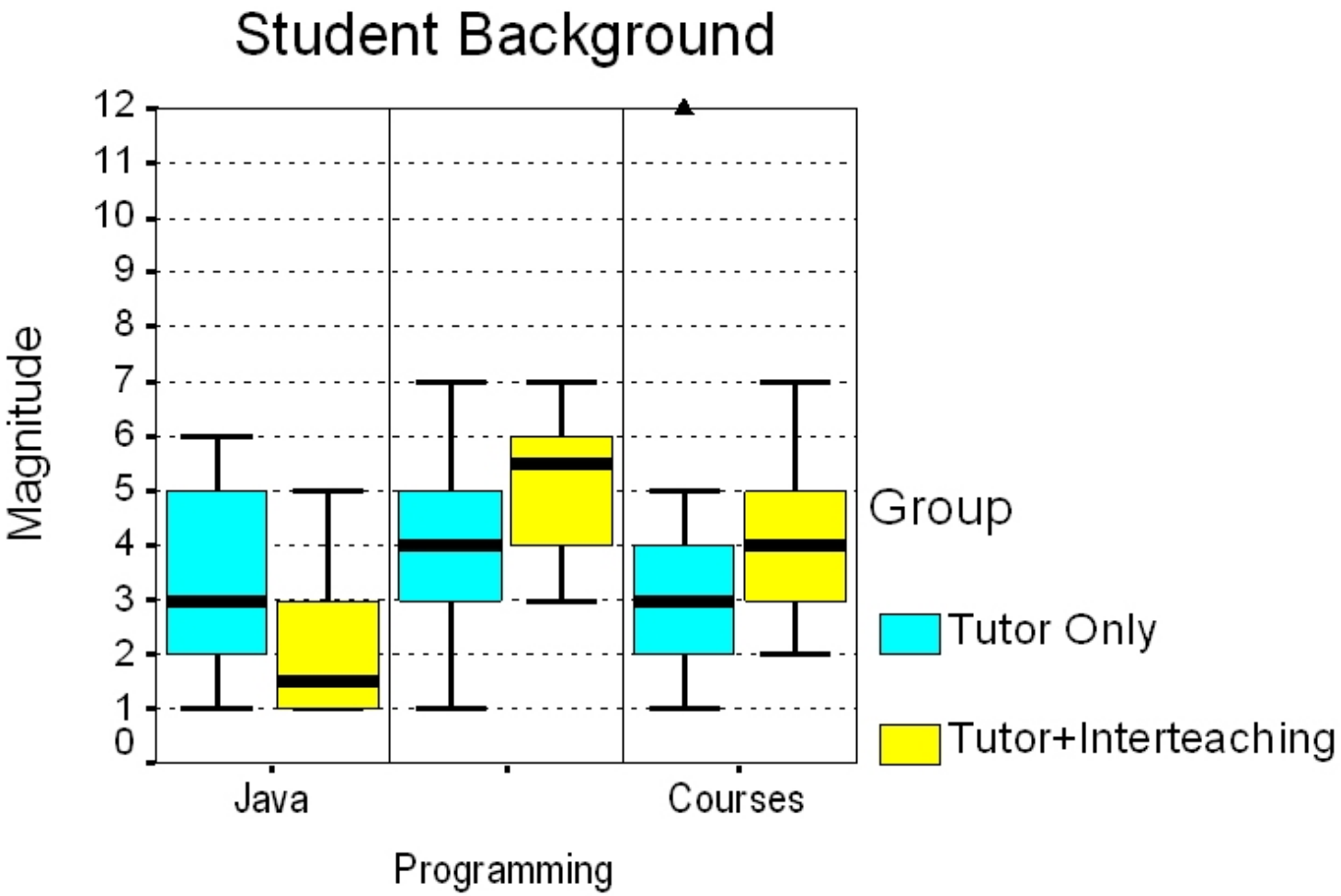
Third Interteaching Report



Interteachers in Action

Classroom Observations

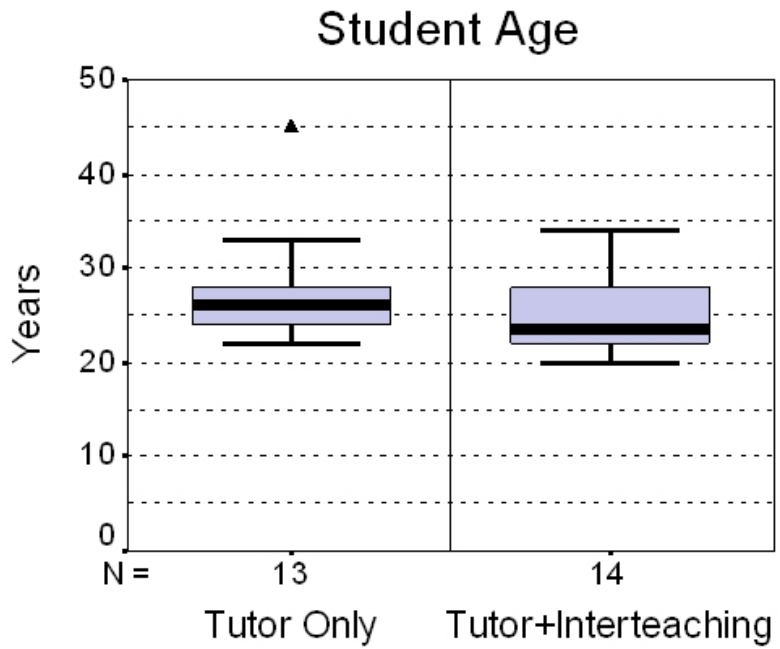
	Summer 2004 (*n* = 14)	Fall 2004 (*n* = 14)
	Tutor Only **Questionnaires: SSE and Rules**	**Tutor + Interteaching** **Questionnaires: SSE, Rules, Java Items, and Rows**
Class 1	• Pre-Tutor Questionnaires • **Tutor** • Post-Tutor Questionnaires	• Pre-Tutor Questionnaires • **Tutor**
		Access to Tutor Study Manual
Class 2	• Lecture • Run the applet • Final Questionnaires	• Post-Tutor Questionnaires • **Interteaching** • Lecture • Run the applet
Class 3		• Final Questionnaires – Test credit

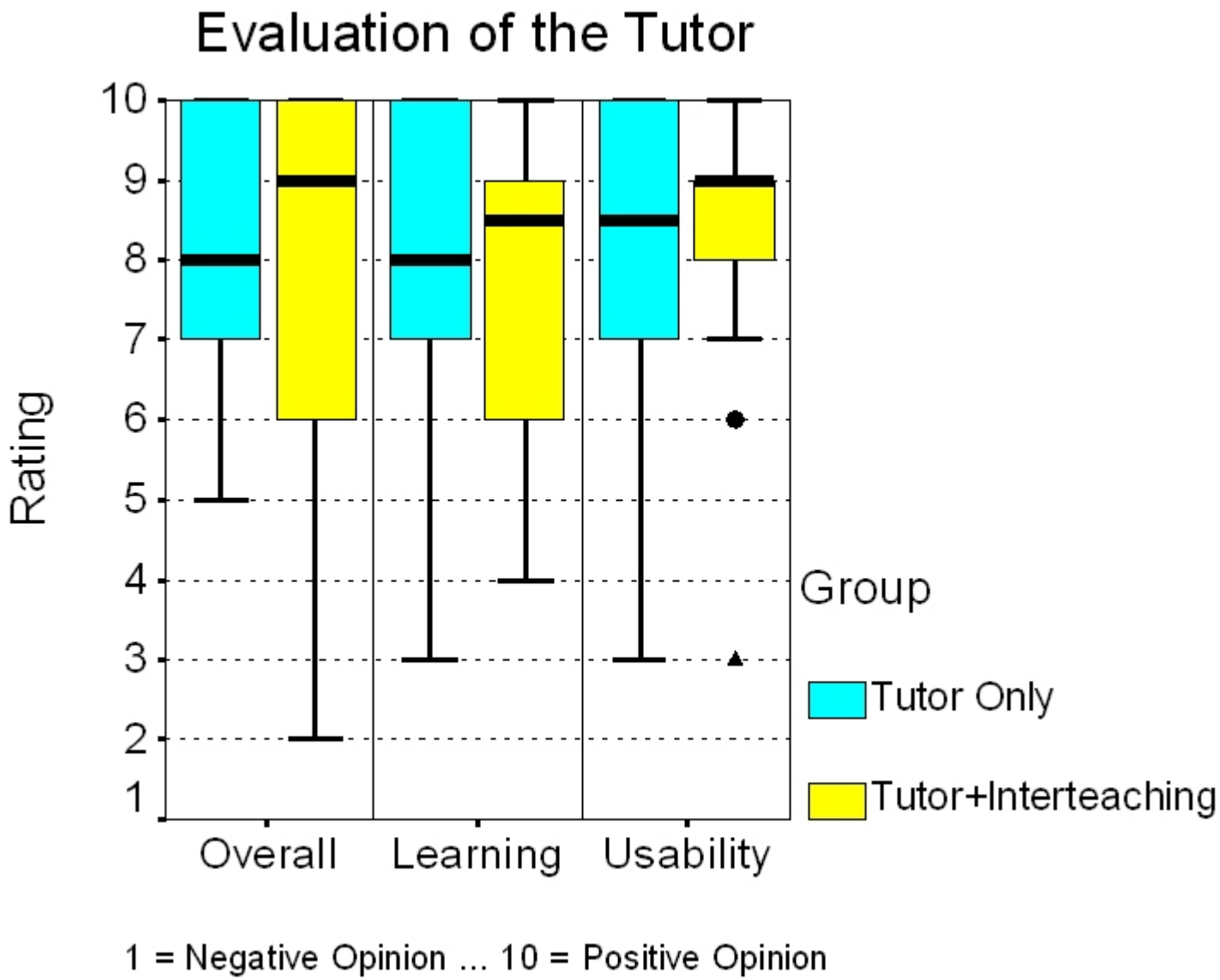


Rated Experience (1 - 10)

Number of Prior Programming Courses (0 - 12)

	Male	Female	Total
Tutor Only	8	6	14
Tutor + IT	13	1	14




Evaluation of the Tutor
Rating
10
9
8
7
6
5
4
3
2
1
Overall
Learning
Usability
Group
Tutor Only
Tutor+Interteaching
1 = Negative Opinion ... 10 = Positive Opinion

Rules Test: 12 Multiple-Choice Questions

11. Which of the following lines would most likely add a JTextField object to a JPanel object?
    a. JPanel.add(JTextField);
    b. JPanel.add(myJTextField);
    c. myJPanel2.add(myJTextField2);
    d. myJPanel.add(JTextField);

Enter a letter here:

How confident are you that you selected the correct answer?

Not at all confident. 1 2 3 4 5 6 7 8 9 10 Totally confident.

Enter a number here:

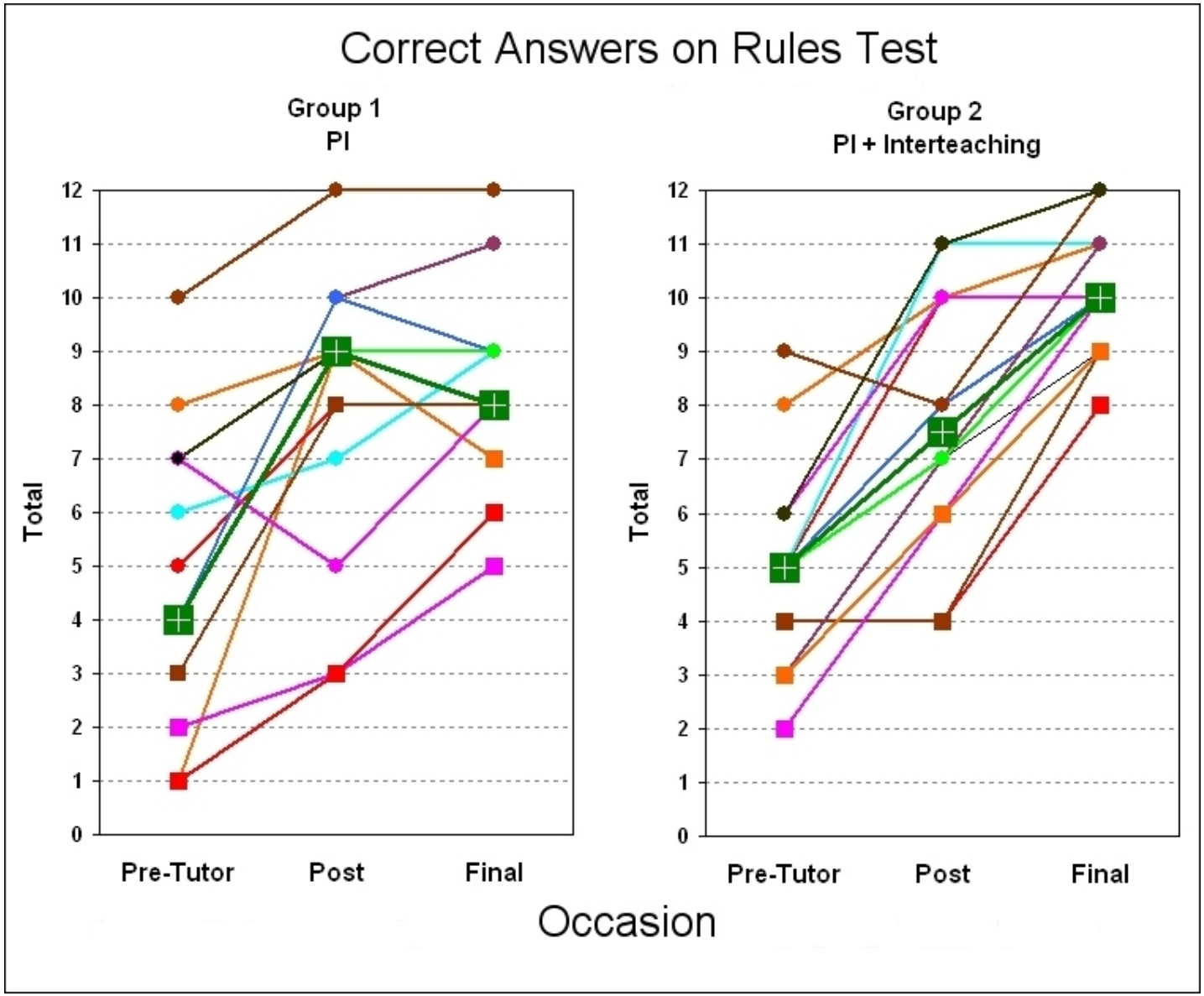
12. Which of the following most likely would be used to change the color of an Applet container?
    a. setBackground(Applet = orange);
    b. this.SetBackground(blue);
    c. SetBackground(Container.red);
    d. this.setBackground(Color.yellow);

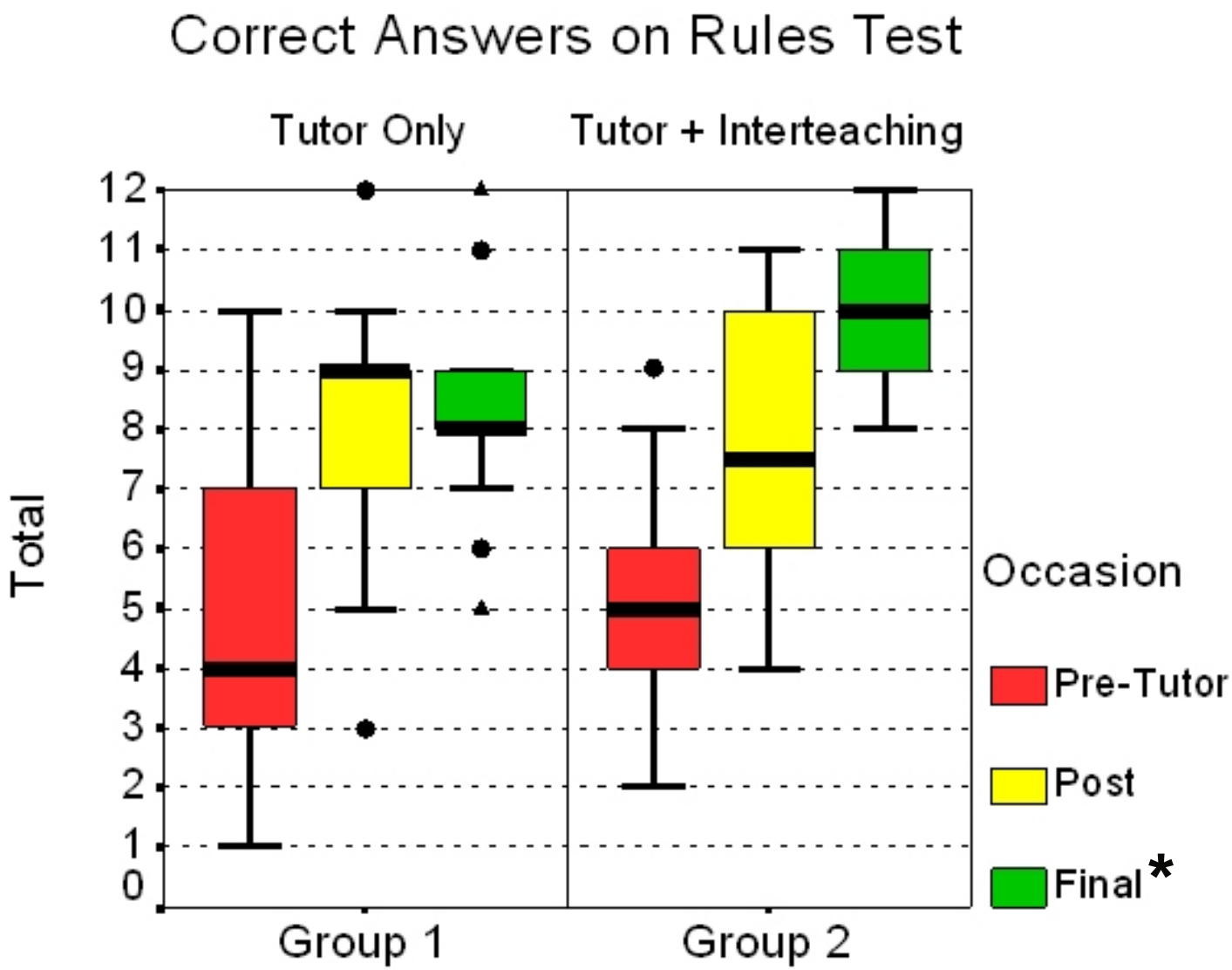
Enter a letter here:

How confident are you that you selected the correct answer?

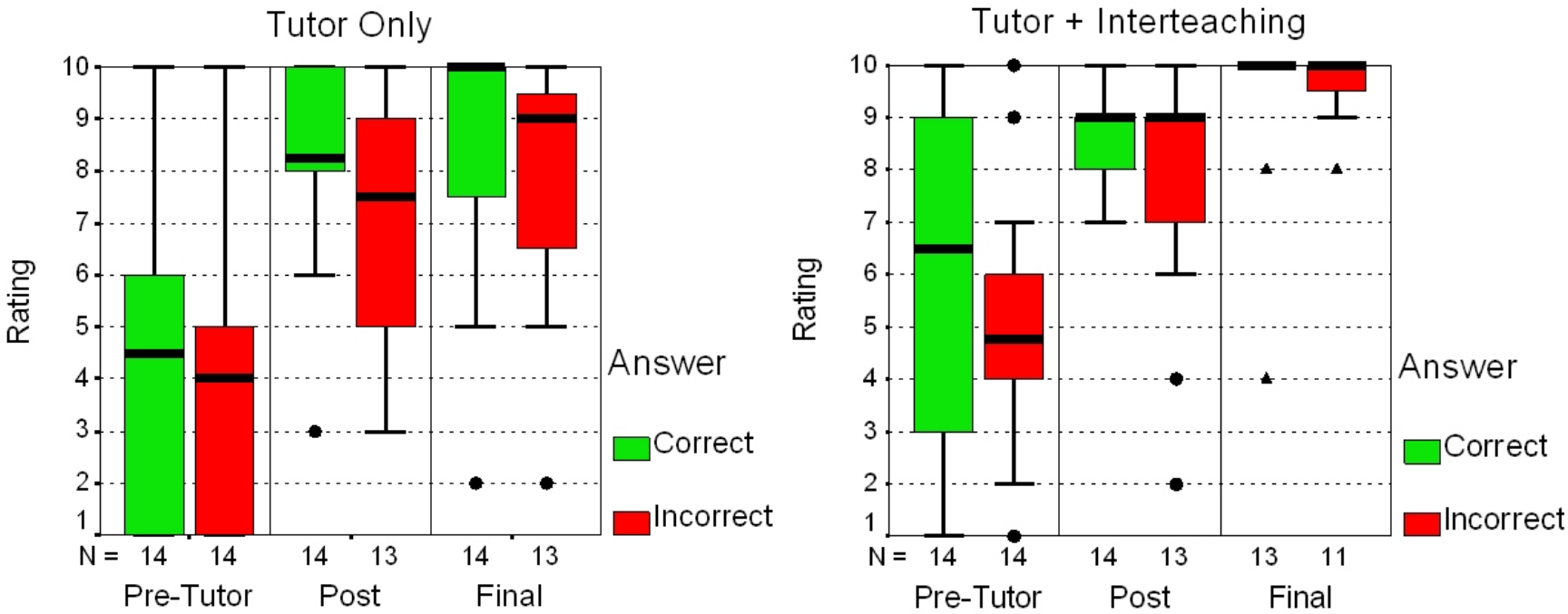
Not at all confident. 1 2 3 4 5 6 7 8 9 10 Totally confident

Enter a number here:


Correct Answers on Rules Test
Group 1
PI
Group 2
PI + Interteaching
Total
12
11
10
9
8
7
6
5
4
3
2
1
0
Pre-Tutor
Post
Final
Occasion


Correct Answers on Rules Test
Tutor Only
Tutor + Interteaching
Total
12
11
10
9
8
7
6
5
4
3
2
1
0
Group 1
Group 2
Occasion
Pre-Tutor
Post
Final *

Confidence in Rules Test Answers



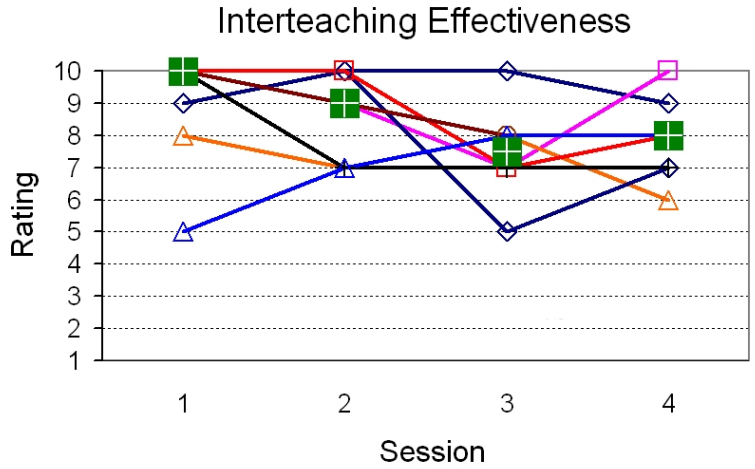
1 = No confidence … 10 = Total confidence

Interteaching Reports

How effective was this session in helping you to learn the material?

1 = Not at all effective. The session did not contribute to my learning of the material.

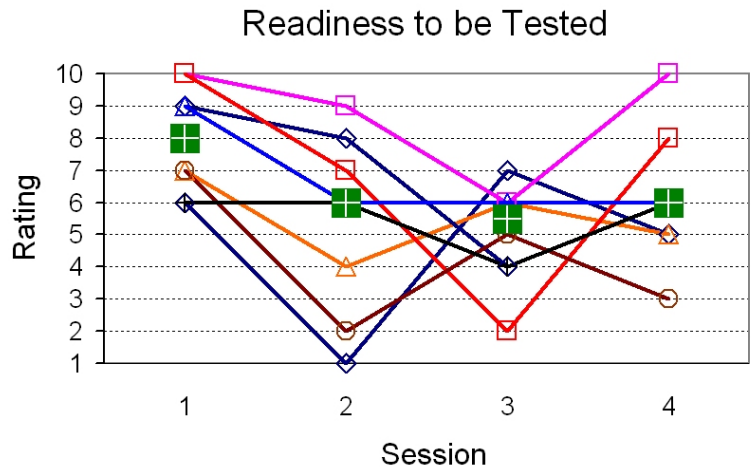
10 = Totally effective. The session contributed to my learning of the material.

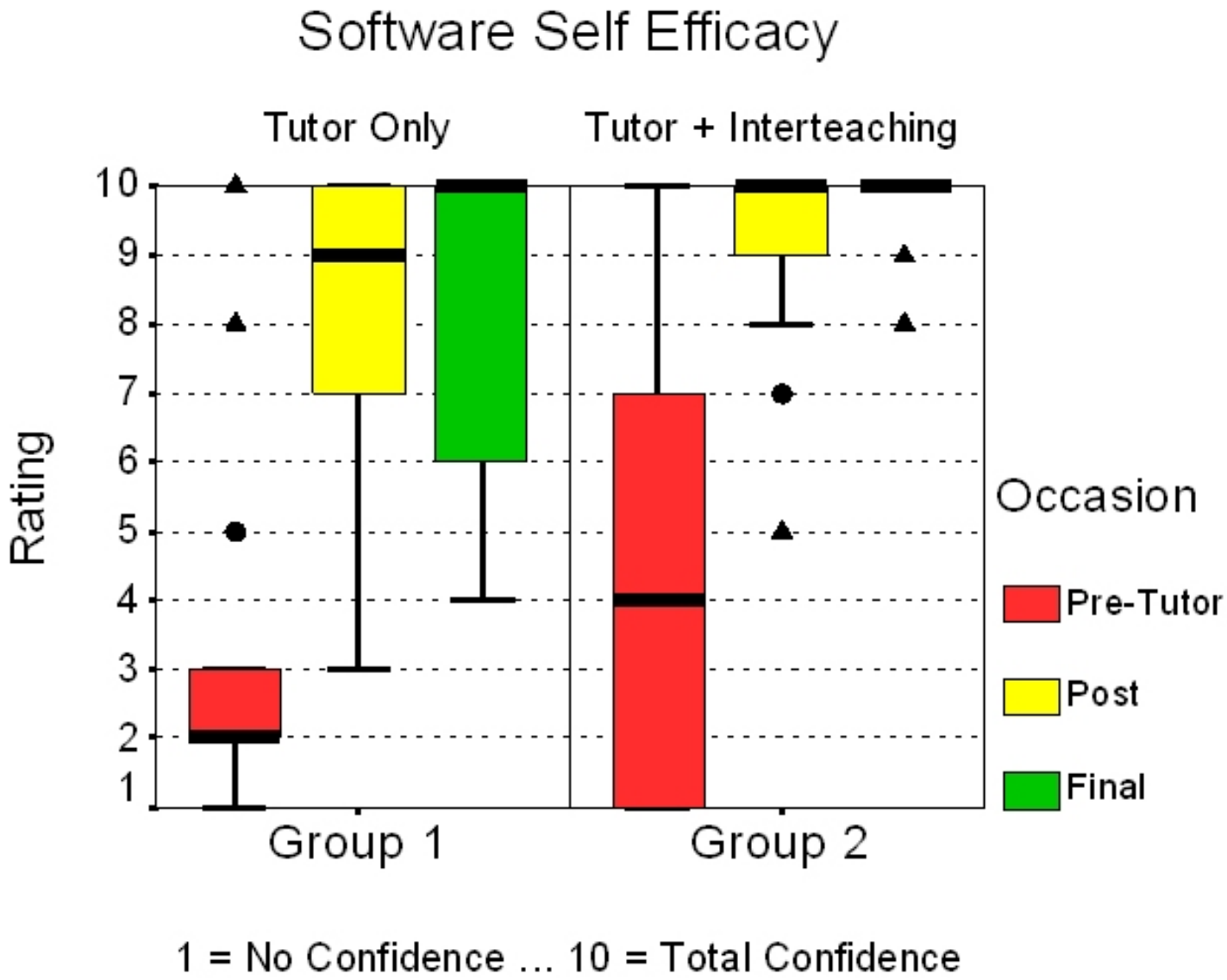


How confident are you that you could answer all questions correctly if you were tested on this program right now?

1 = Not at all confident. I could not answer any question correctly.

10 = Totally confident. I could answer all the questions correctly.

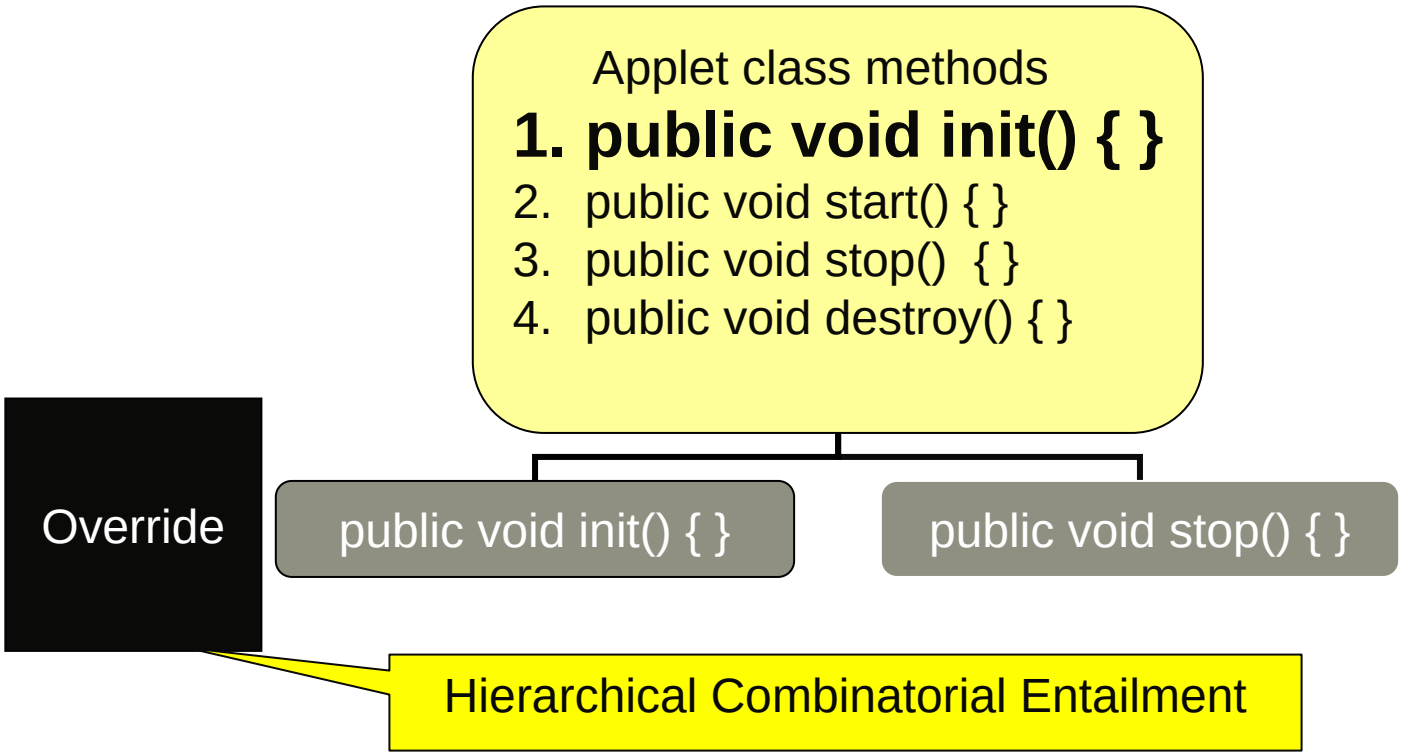



Software Self Efficacy
Tutor Only
Tutor + Interteaching
Rating
10
9
8
7
6
5
4
3
2
1
Group 1
Group 2
Occasion
Pre-Tutor
Post
Final
1 = No Confidence … 10 = Total Confidence

Equivalences

- public void init() {} == public void stop() {}
- How was **public void stop() {}** recognized as a valid form for a method when that particular form did not appear in the tutor?

Relationally Framing



Challenges with Programmed Instruction

- It is labor intensive to develop.
  - We have proposed to develop a generic shell.
- There are conceptual issues regarding the size of a learn unit.
  - The opportunity for repetition can lead to careless reading.

Conclusions

1. Programmed instruction is an effective tool in technology education.
   - It meets the needs of the individual learner.
   - The instructional design can promote meaningful learning and self-confidence.
   - The tutoring system is well-received by novitiate learners.
2. Interteaching may add value.
3. The competency attained sets the occasion for advanced learning with enthusiasm.
4. Students like the tutor and the interteaching, and so do I.

Thank you!

Questions?

http://nasa1.ifsm.umbc.edu/