

Combining Programmed Instruction and Collaborative Peer Tutoring to Teach Java™



Henry H. Emurian

Information Systems Department

College of Engineering and Information Technology

UMBC

1000 Hilltop Circle

Baltimore, Maryland 21250

What I do at UMBC:

- Among other things, I teach Java to Information Systems (IS) majors.

- An ***optimal teaching strategy*** in the 21st century should be one that respects the right of each and every student to have the opportunity to achieve mastery, where opportunity is taken to mean unlimited exposure to the proper conditions of learning until an achievement outcome has been attained.

Forward and Forewarned

- ***As stated by Anderson et al. (1995), It is more meaningful to hold constant the level of mastery required and look at differences in time to achieve that level. This reflects the true gain of an educational technique (p. 185).***

- We aim for that **true gain** by our students in response to our instructional tactics in technology education.

- Observe students in context
 - Repeated observations in one classroom with one group of students and one instructor
- Improve the instructional design over successive replications
- Emphasizes movement of all students to a common learning outcome (*True Gain*)
 - Contrasts with between-group studies concerned with effect size differences

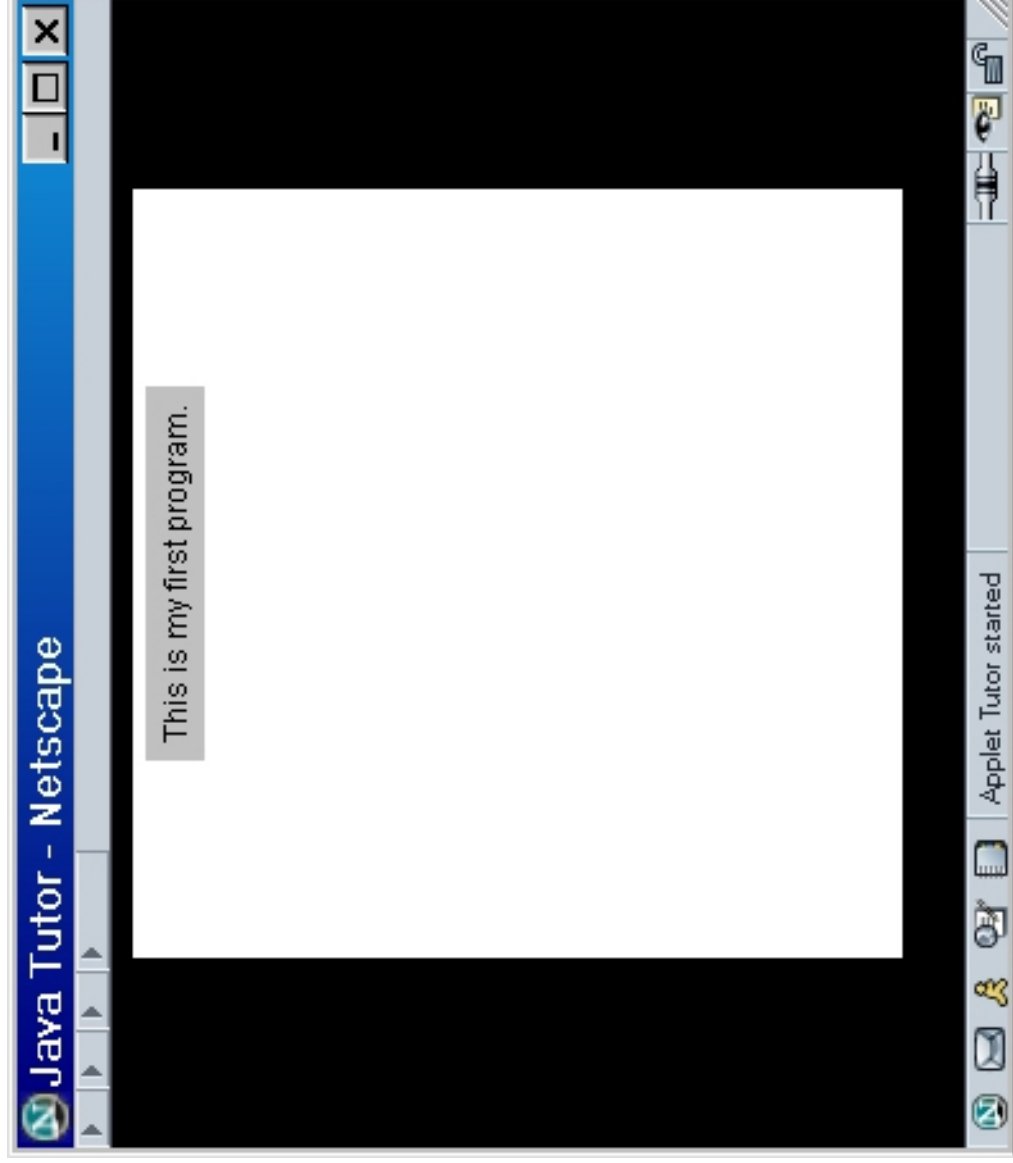
Performance

```
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. }
```

•Near transfer (understand and recite)

•Far transfer (meaningful learning → solve novel problems)

Consequence

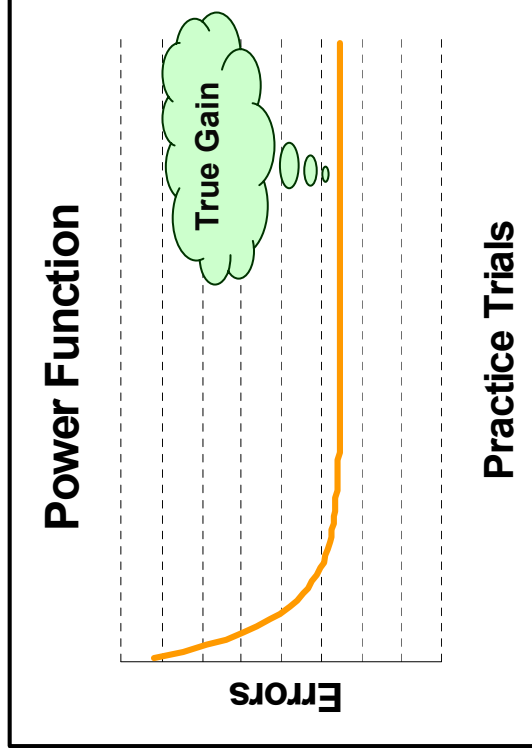


Challenges

- 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 best **help** IS students achieve the objective?
 - **Programmed instruction + Lecture + Collaborative Peer Learning**

Model

- Principles to promote retention and transfer:
 - **Repeated practice** with different instructional modalities (Halpern & Hakerl, 2003)
 - **Socially supported interactions** (Fox & Hackerman, 2003)



What instructional modalities make sense?

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 facts to the achievement of a meaningful learning.



1. **Lecture**
 - Repeat the tutor material while students write the code
2. Run the applet on the web

Far Transfer Miseries

Programmed
Instruction

What I got...

Lecture

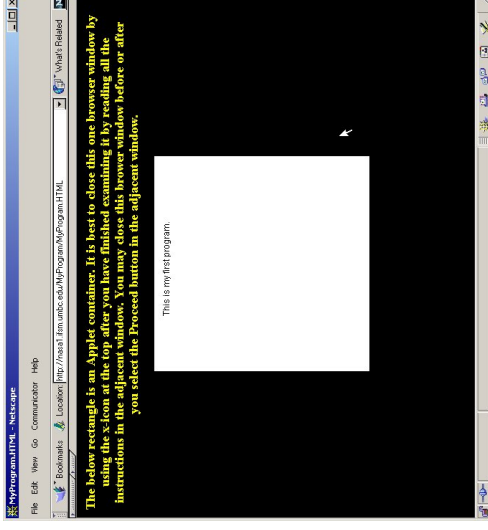
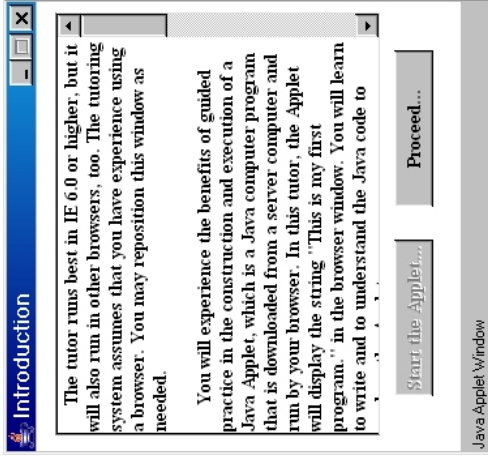
Interteaching



1. A **mutually probing, mutually informing conversation between two people** (Boyce & Hinline, 2002)
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

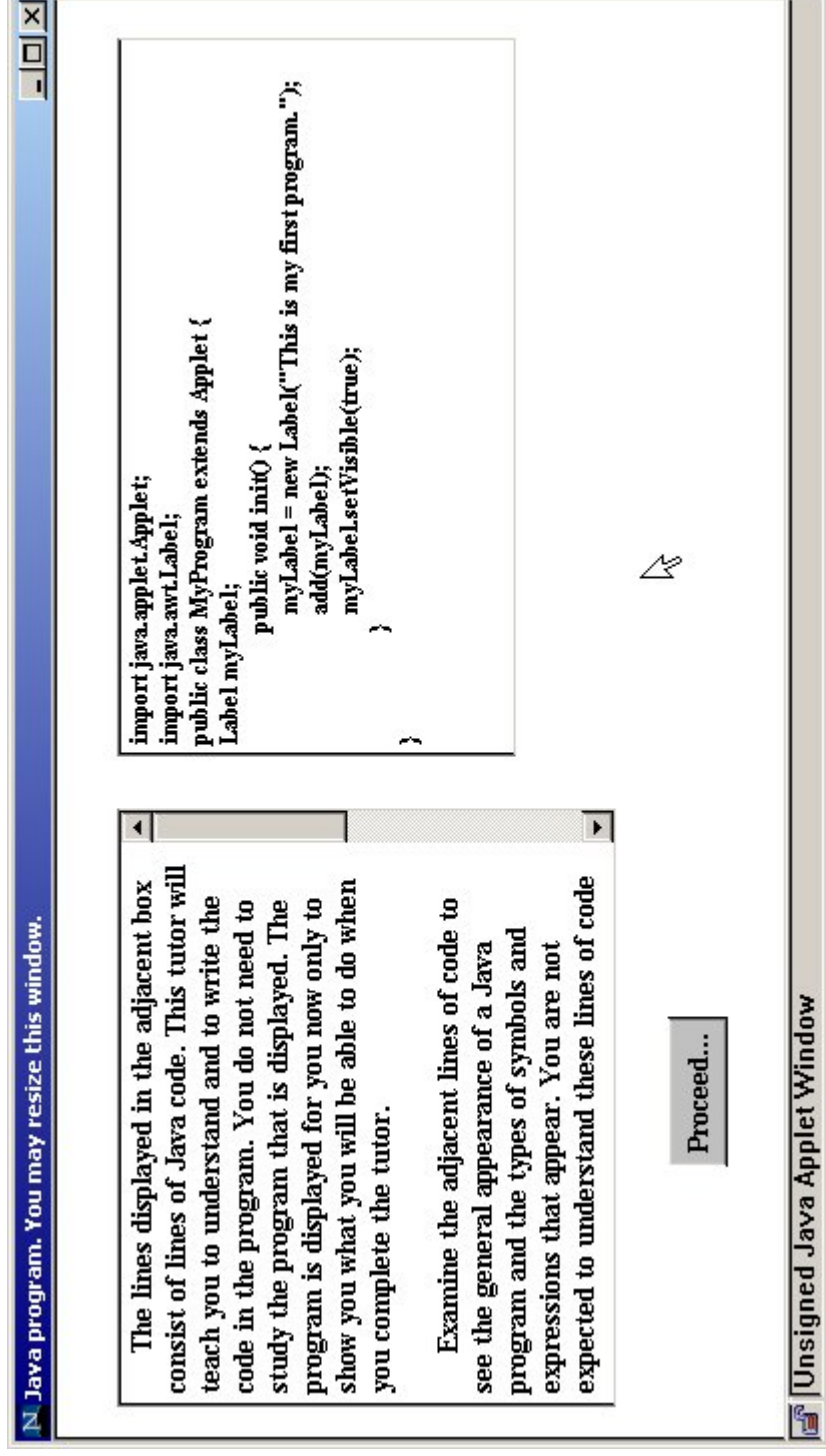
Programmed Instruction Tutoring System

Introduction



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

Applet Code Orientation



The screenshot shows a window titled "Unsigned Java Applet Window". At the top, a blue bar contains the text "Java program. You may resize this window." Below this, there are two main sections. On the left, a text box contains instructions: "The lines displayed in the adjacent box consist of lines of Java code. This tutor will teach you to understand and to write the code in the program. You do not need to study the program that is displayed. The program is displayed for you now only to show you what you will be able to do when you complete the tutor." Below this text is a "Proceed..." button. On the right, a code editor displays the following Java code:

```
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);
    }
}
```

A mouse cursor is visible over the code editor. At the bottom of the window, the title bar reads "Unsigned Java Applet Window".

HTML Orientation

The screenshot shows a Java Applet window titled "Unsigned Java Applet Window". The window contains a text area with the following HTML code:

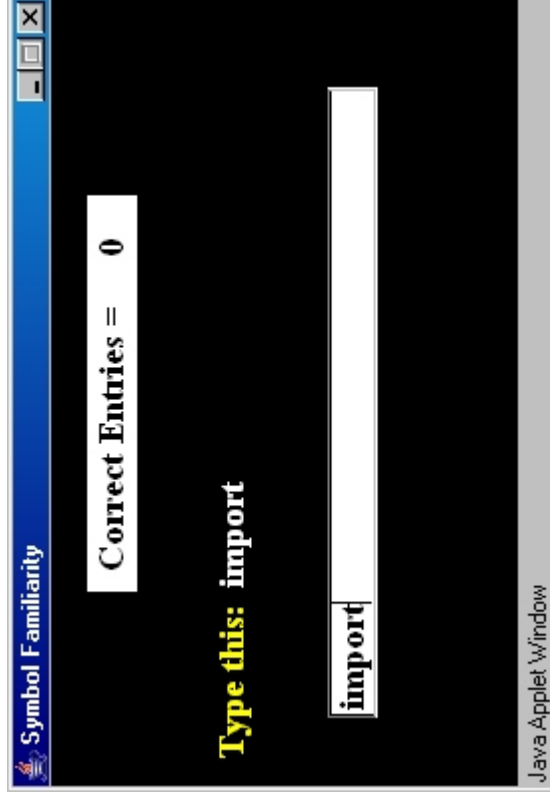
```
<HTML>
<TITLE>MyProgram.HTML</TITLE>
<BODY BGCOLOR=black>
<CENTER>
<APPLET CODE = "MyProgram.class" HEIGHT = 300 WIDTH = 300>
</APPLET>
</CENTER>
</BODY>
</HTML>
```

Below the code, there is a "Proceed..." button. The window also contains instructional text:

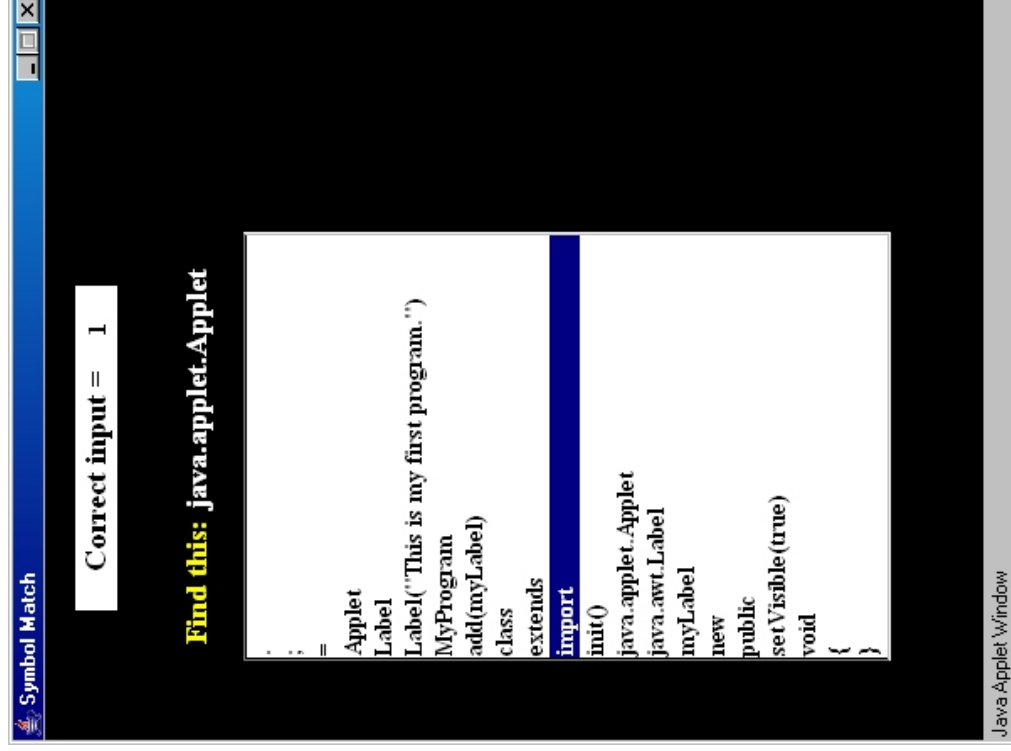
The lines displayed in the adjacent box consist of HTML tags and parameters to run the MyProgram.class program, which is produced by compiling the Java code. The lines are created with a text editor and saved as MyProgram.HTML. There is no compilation with the HTML file. It is used as it was written in the editor.

The Java class file, which is executed as an Applet, is started by using MyProgram.HTML as the target file in the browser URL.

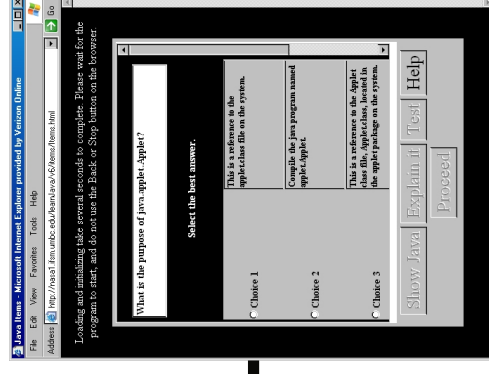
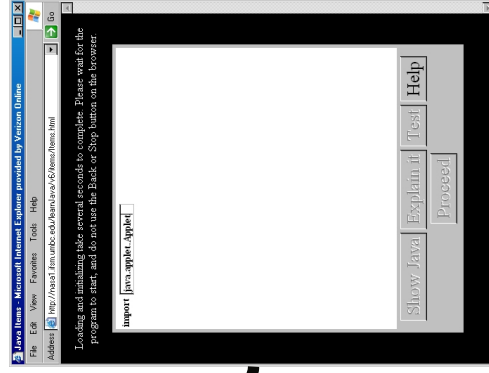
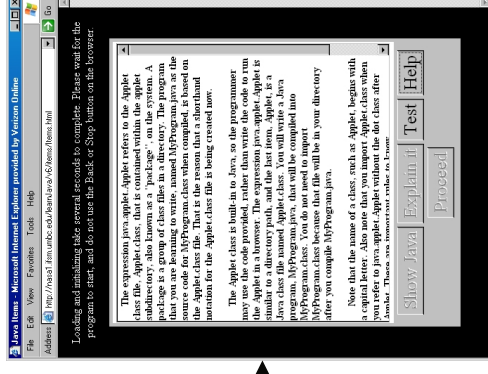
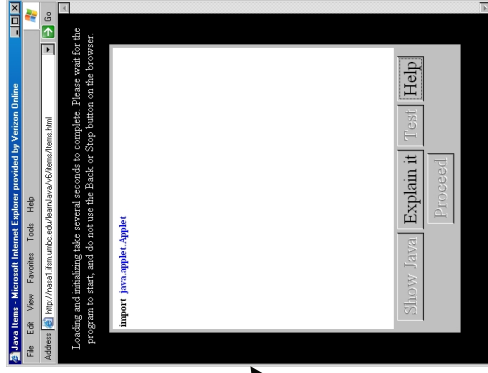
Textual Imitation



Discrimination Training



Item Learning



Input Field

```
import
```

```
import java.applet.Applet;
```

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

Item Rule Example 1

Rules

`.Applet.class` file. That is the reason that a shorthand notation for the `Applet.class` file is being created now.

The `Applet` class is built-in to Java, so the programmer may use the code provided, rather than write all the code to run the `Applet` in a browser. Some of the code that you need has already been written for you. The expression `'java.applet.Applet'` is similar to a directory path, and the last item, `Applet`, is a Java class file named `Applet.class`. You will write a Java program, `MyProgram.java`, that will be compiled into `MyProgram.class`. You do not need to import `MyProgram.class` because that file will be in your directory after you compile `MyProgram.java`.

Note that the name of a class, such as `Applet`, begins with a capital letter. Also note that you import `Applet.class` when you refer to `'java.applet.Applet'` without the dot class after `.Applet`. These are important rules to know.

After you use `'import java.applet.Applet;'` at the beginning of your program, you can use `Applet` by itself anywhere in the program, and the compiler will know where to find the `Applet.class` file on the system.

Show Java

Explain it

Test

Help

Proceed

Item Rule Example 2

The ';' mark (semi-colon) designates the end of a series of Java terms that are compiled as a single unit or statement or line. It has the function of an end-of-line marker. The ';' mark is known as a separator term in the Java programming language. It separates one statement, or line, from the next.

The code in row 1, then, is as follows:

```
'import java.applet.Applet;'
```

Look carefully at the code, and rehearse in your mind your current understanding of the code. You will overcome any misunderstanding later. Do notice that the 'import' keyword is followed by a directory path that has the name of a Java 'class' as the last member in the path. This is an important rule to know.

Rule

Test Example

Which of the following could be the last in a path used with 'import java.awt.'?

Select the best answer.

Choice 1

Choice 2

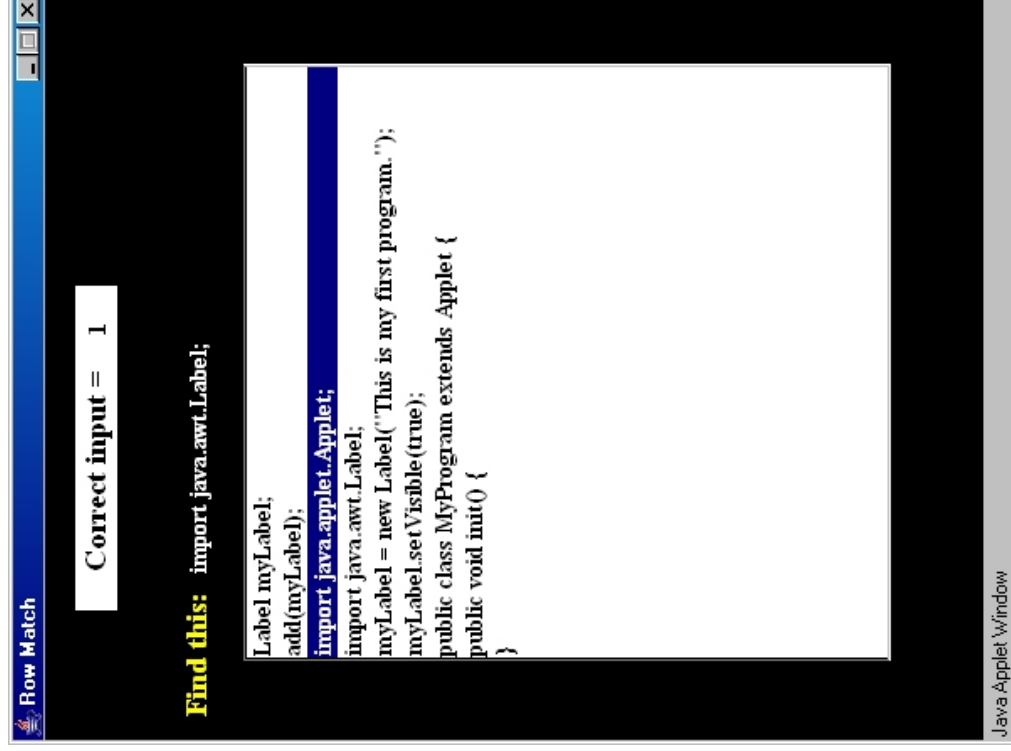
Choice 3

java.applet
TextField
Applet

Row Imitation



Row Discrimination Training



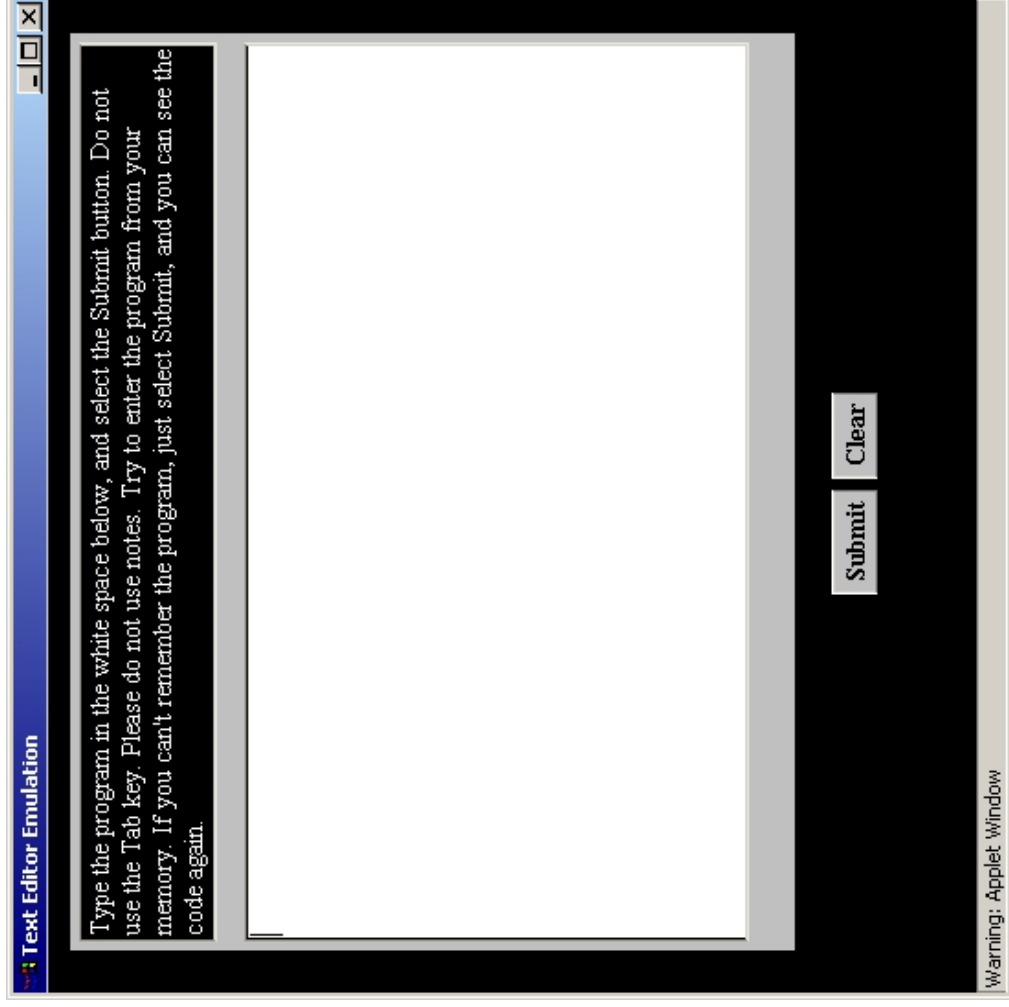
Row Learning

The screenshot shows a window titled "Java Tutoring System: Pass 1 of 2". Inside the window, there is a code editor with 10 rows. The first two rows contain Java code: "import java.applet.Applet;" and "import java.awt.Label;". The remaining eight rows are empty. To the right of the code editor, there is a question: "Explain the code?". The window title bar at the bottom right reads "Java Applet Window".

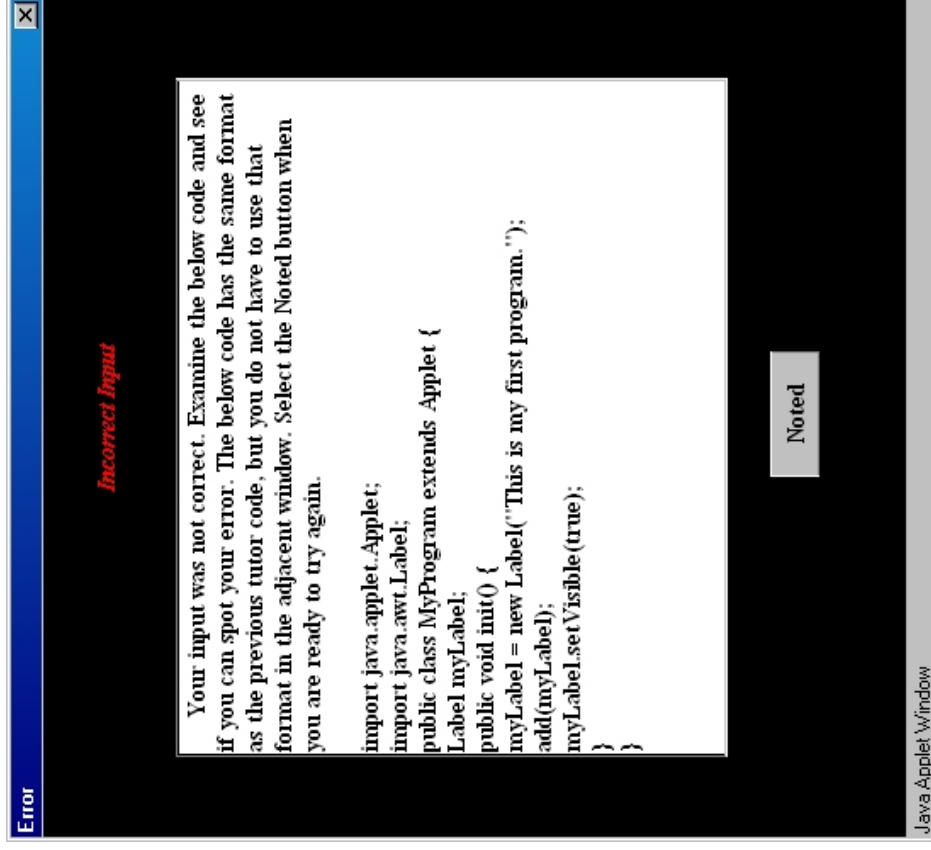
Row	Code
Row 1	import java.applet.Applet;
Row 2	import java.awt.Label;
Row 3	
Row 4	
Row 5	
Row 6	
Row 7	
Row 8	
Row 9	
Row 10	

Explain the code?

Program Interface



Program Interface



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?

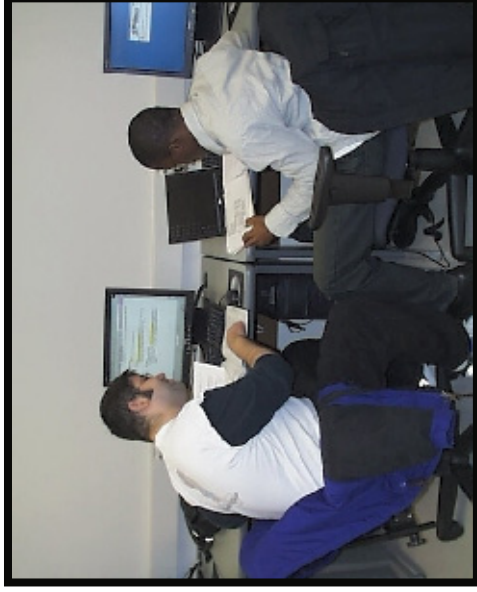
1 = Not at all confident. I could not answer any question correctly.

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**.

Interteachers in Action



Classroom Observations

Summer 2004 ($n = 14$)

Fall 2004 ($n = 14$)

2.5 hours	Tutor	Tutor + Interteaching
Class 1	<ul style="list-style-type: none"> • Pre-Tutor Questionnaires • Tutor • Post-Tutor Questionnaires 	<ul style="list-style-type: none"> • Pre-Tutor Questionnaires • Tutor
		Access to <i>Tutor Study Manual</i>
Class 2	<ul style="list-style-type: none"> • Lecture • Run the applet • Final Questionnaires 	<ul style="list-style-type: none"> • Post-Tutor Questionnaires • Interteaching • Lecture • Run the applet
Class 3		<ul style="list-style-type: none"> • Final Questionnaires – Test credit

Software Self-Efficacy Ratings: 21 Items

Question 4

How confident are you that you can use the following symbol now to write a Java program?

Applet

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

Enter a number here:

Question 5

How confident are you that you can use the following symbol now to write a Java program?

Label

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

Enter a number here:

Question 6

How confident are you that you can use the following symbol now to write a Java program?

MyProgram

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

Enter a number here:

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);`

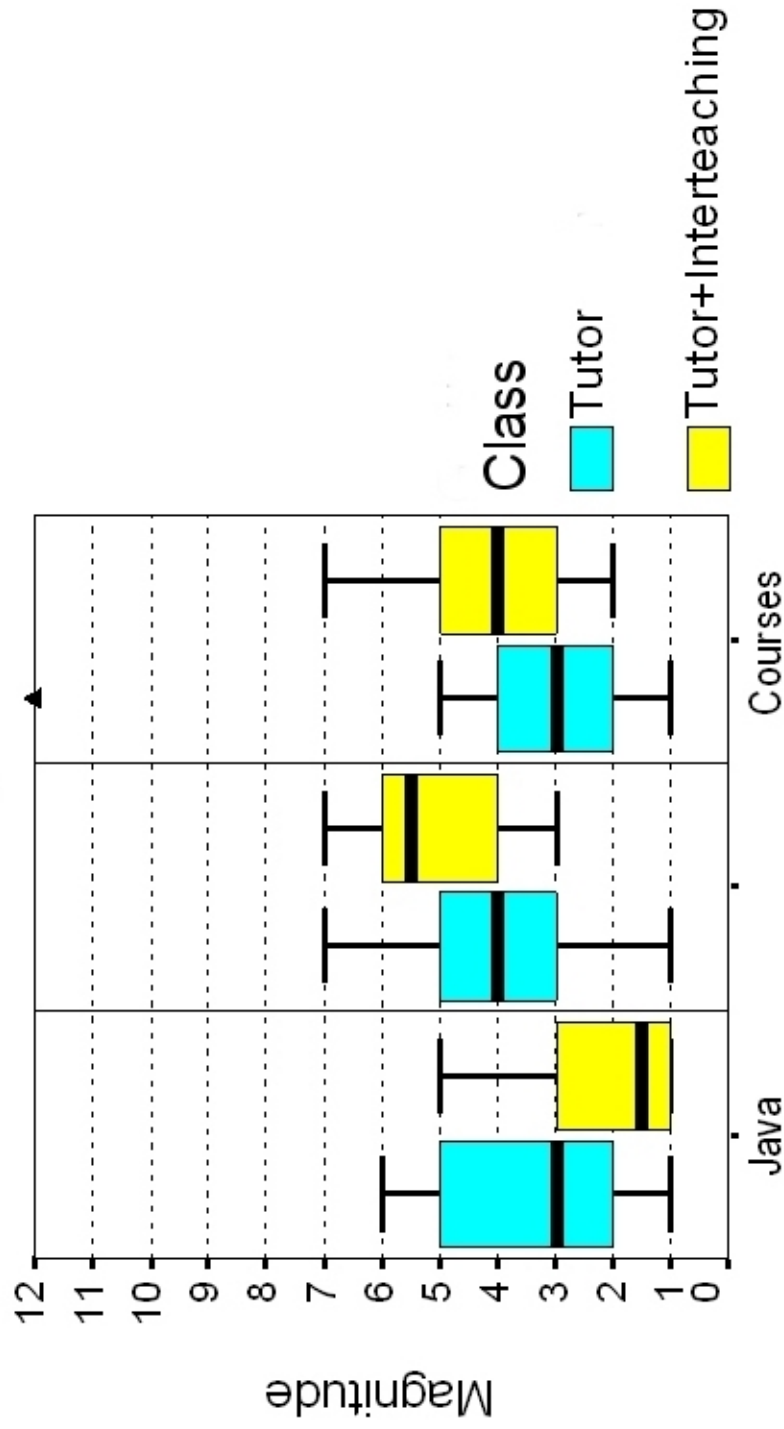
Enter a letter here:

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

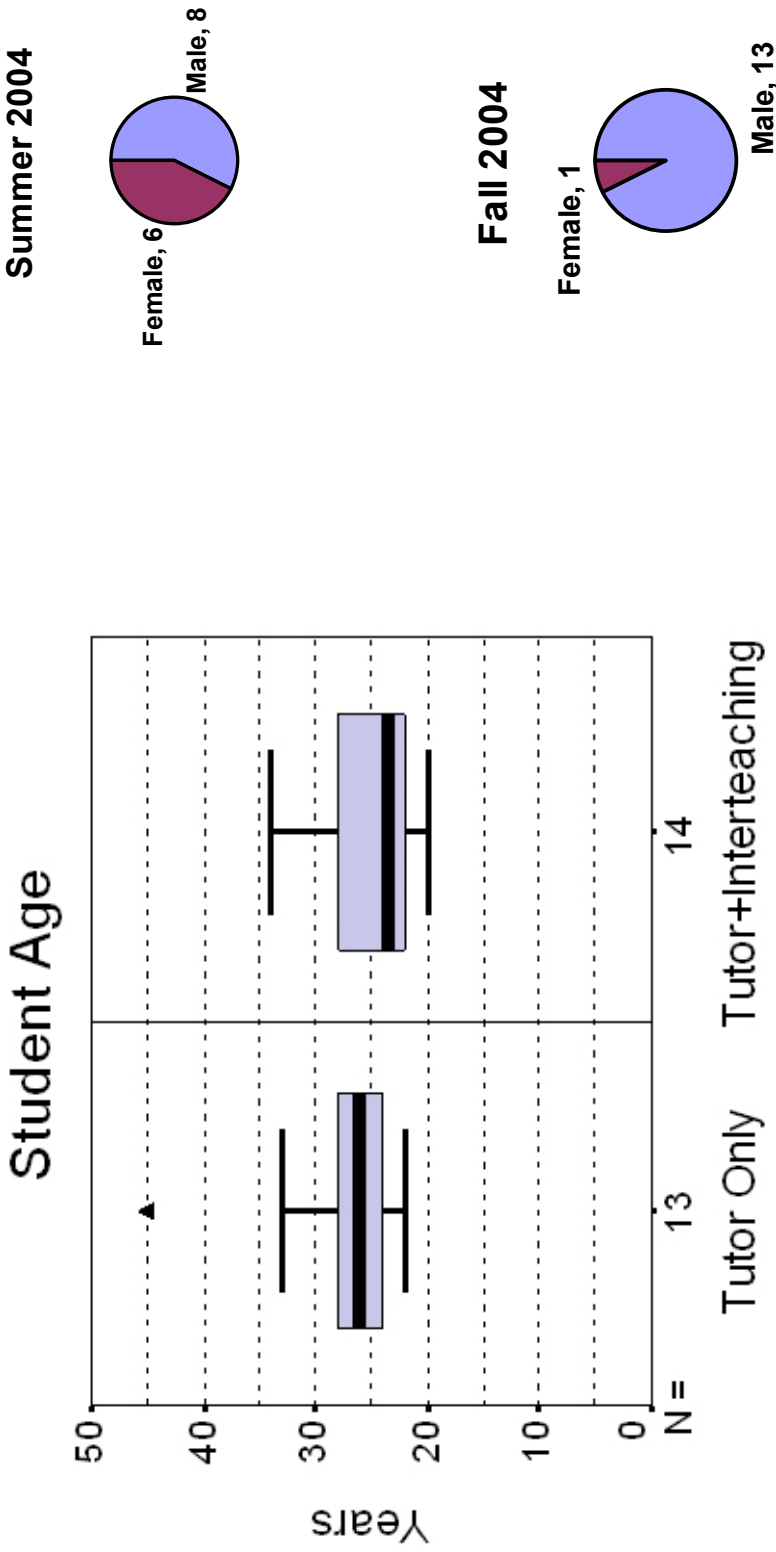
Enter a number here:

Student Background

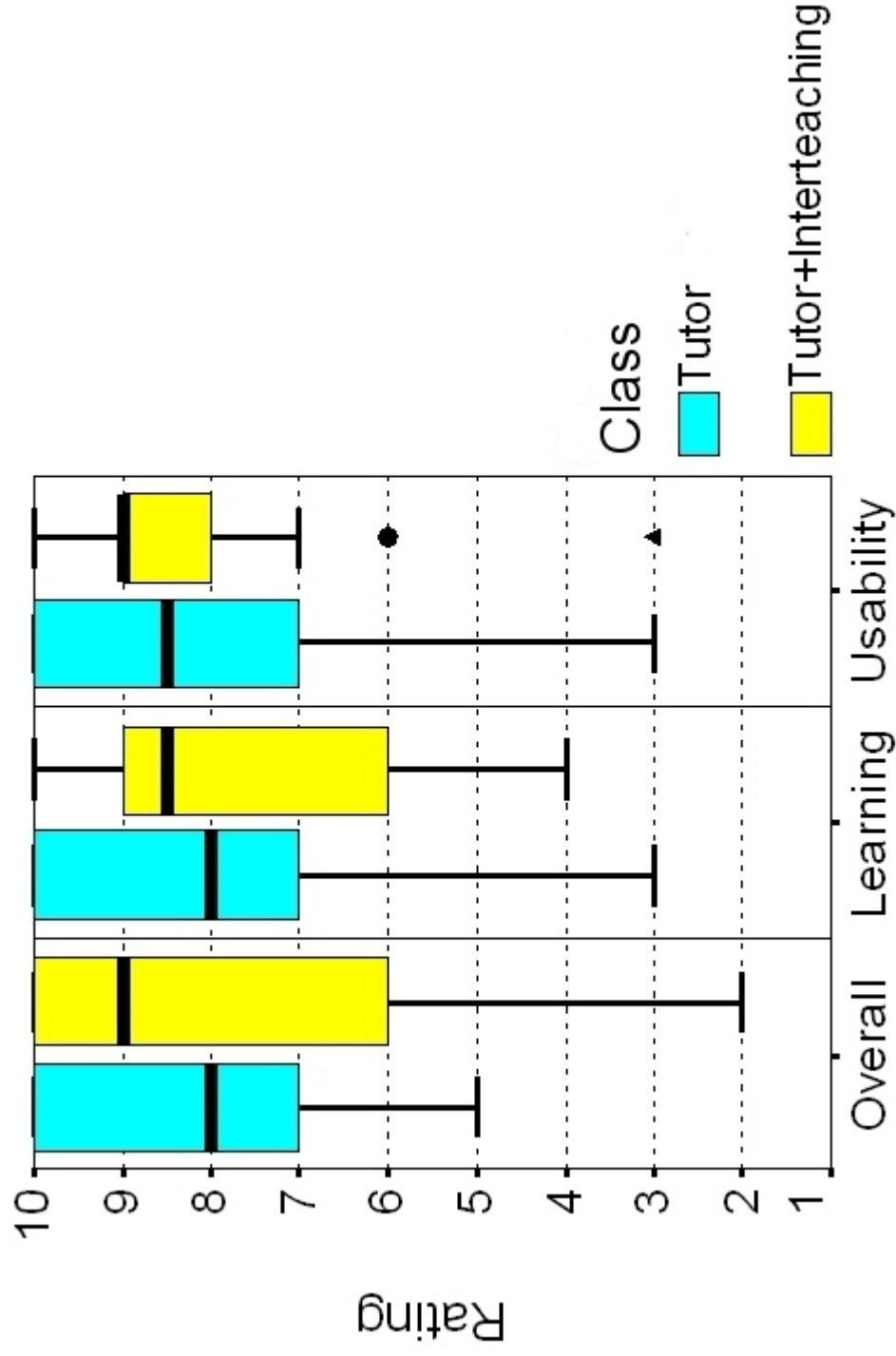


Rated Experience (1 - 10)

Number of Prior Programming Courses (0 - 12)

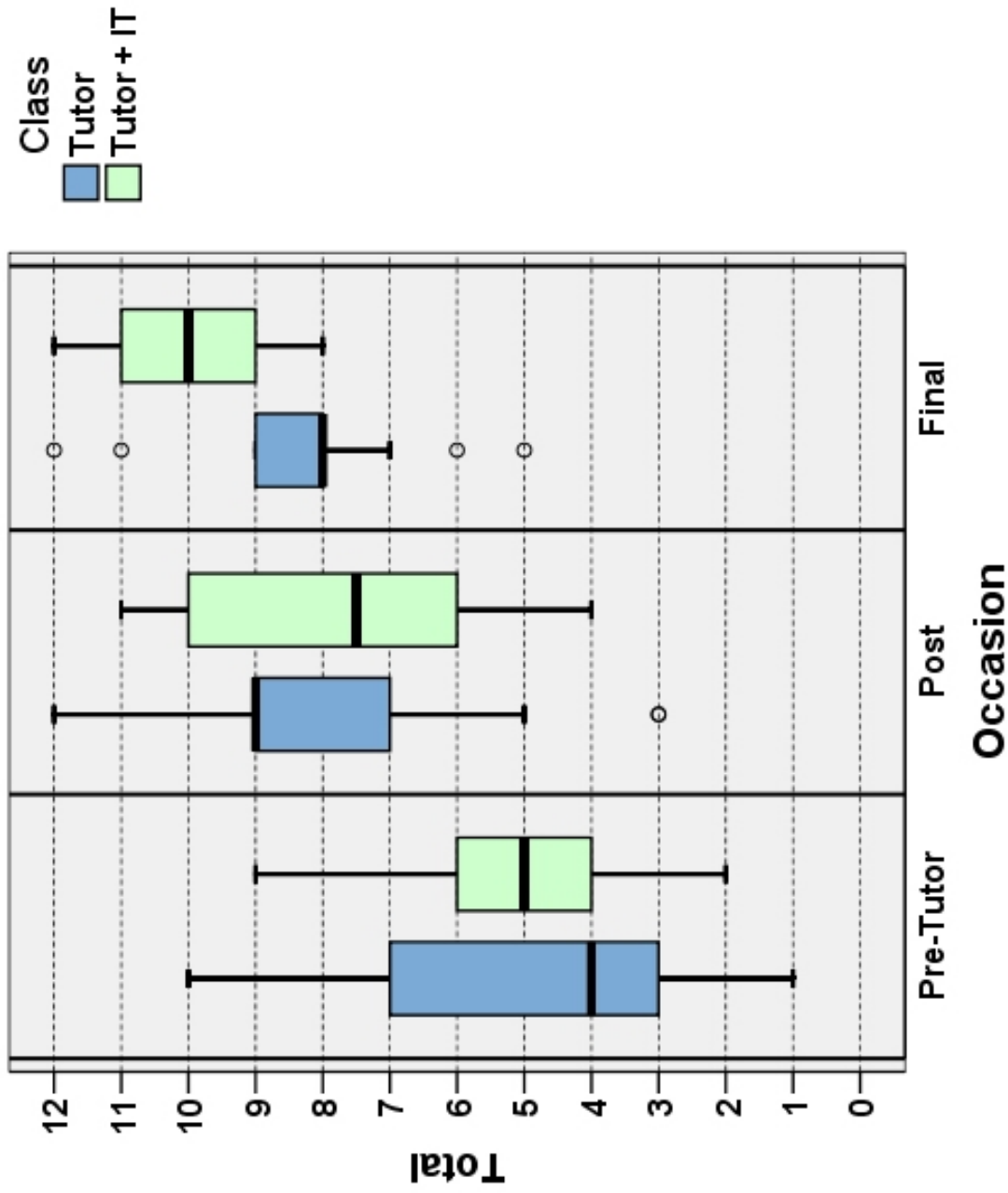


Evaluation of the Tutor



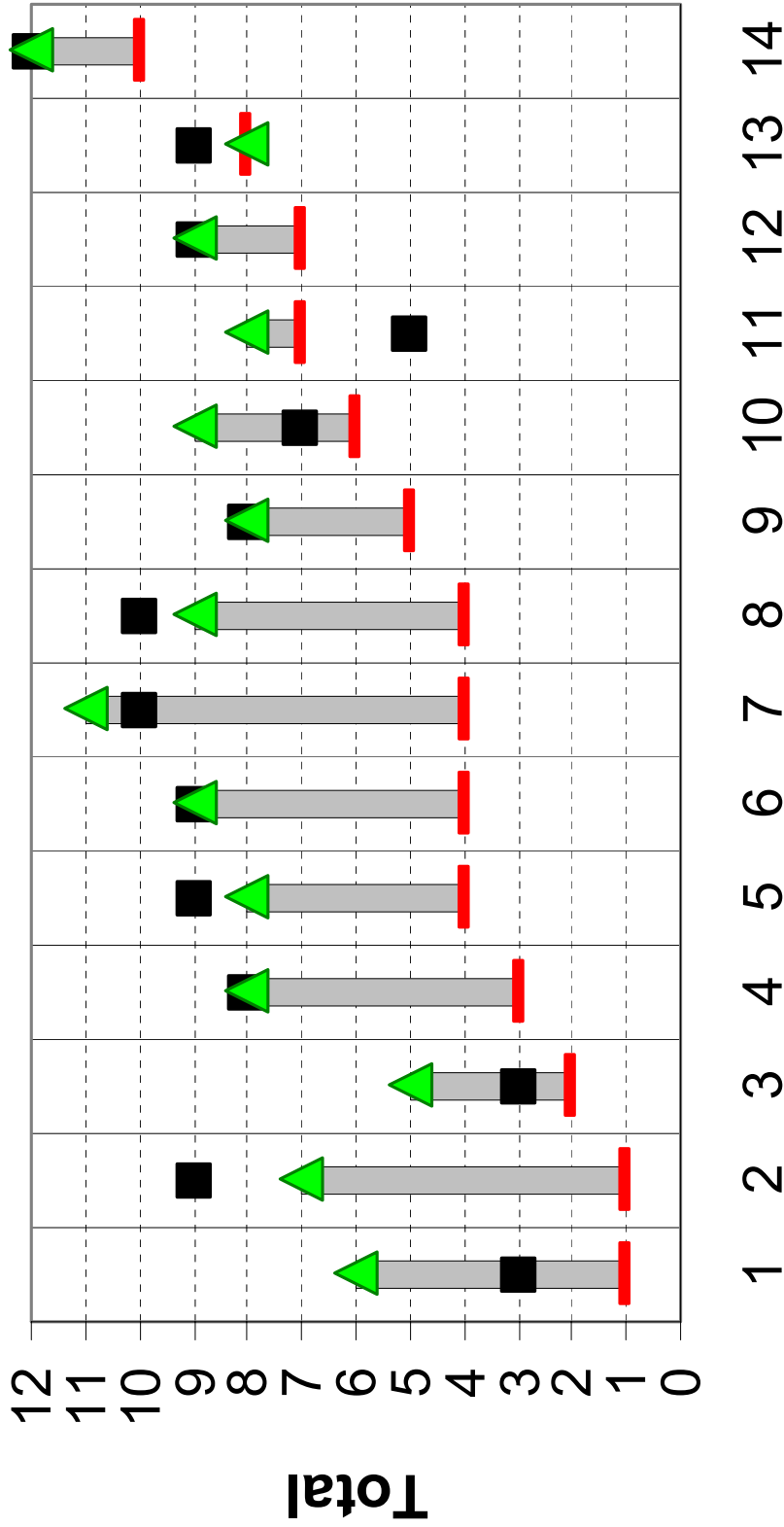
1 = Negative Opinion ... 10 = Positive Opinion

Correct Answers on Rules Test



Correct Rules Test Answers

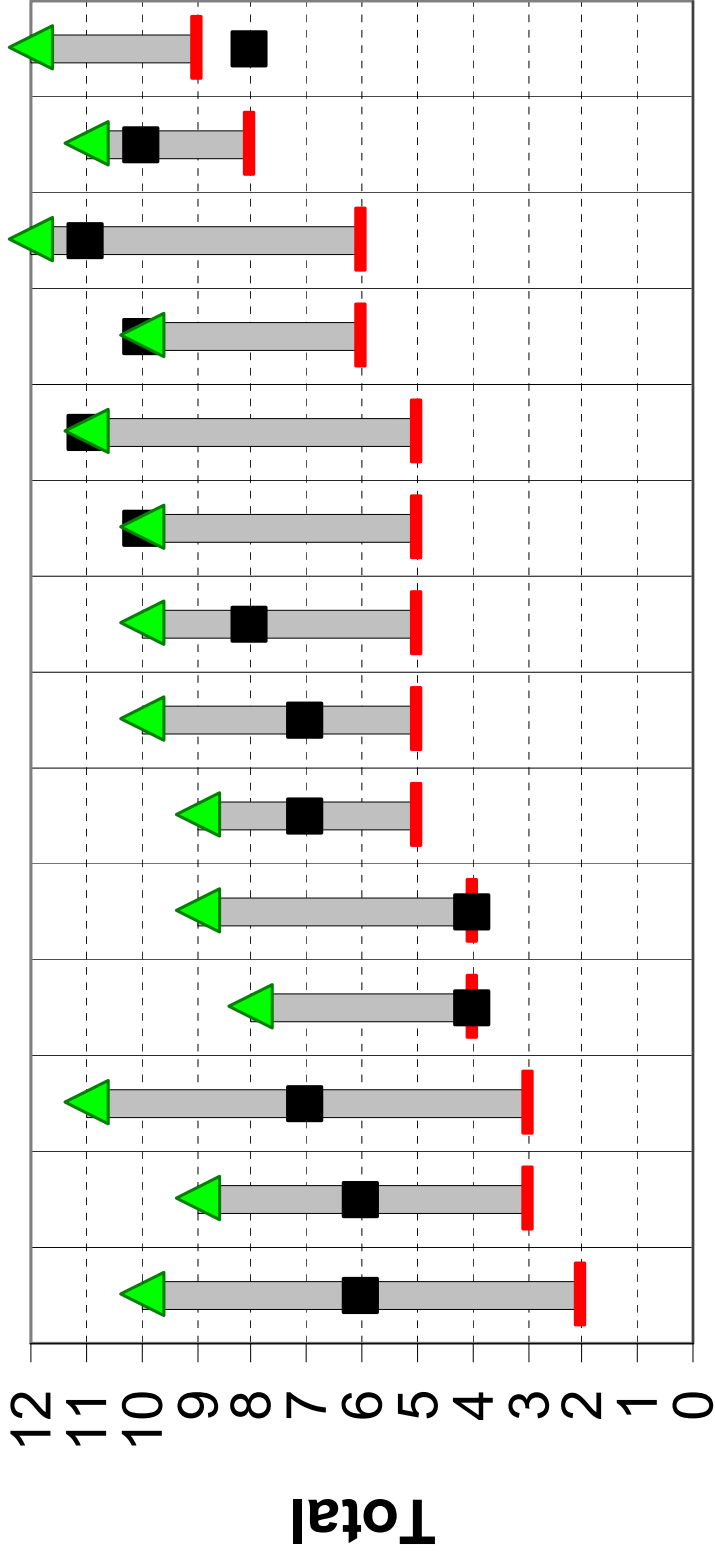
— Pre-Tutor ■ Post-Tutor ▲ Lecture



Tutor Class Members

Correct Rules Test Answers

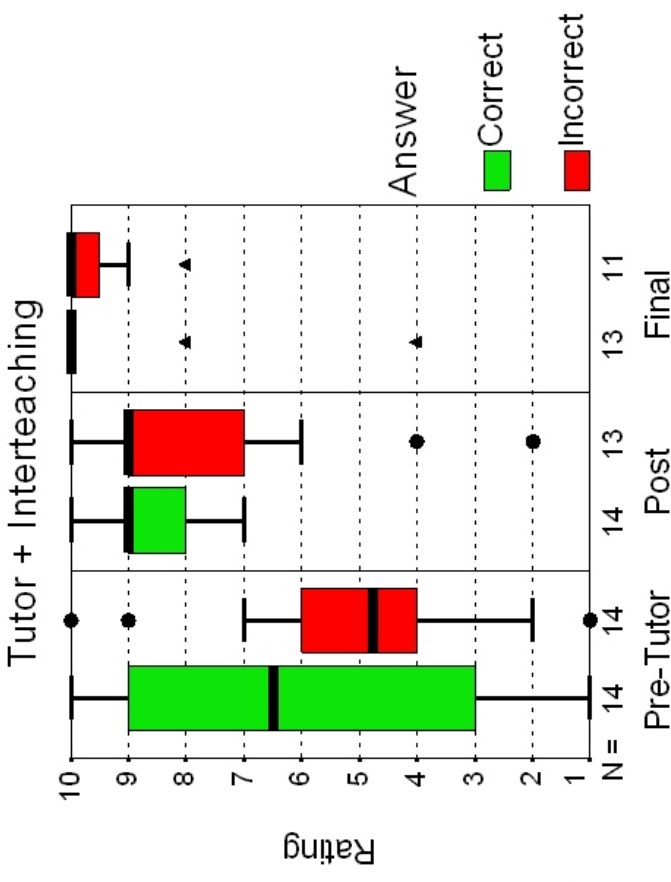
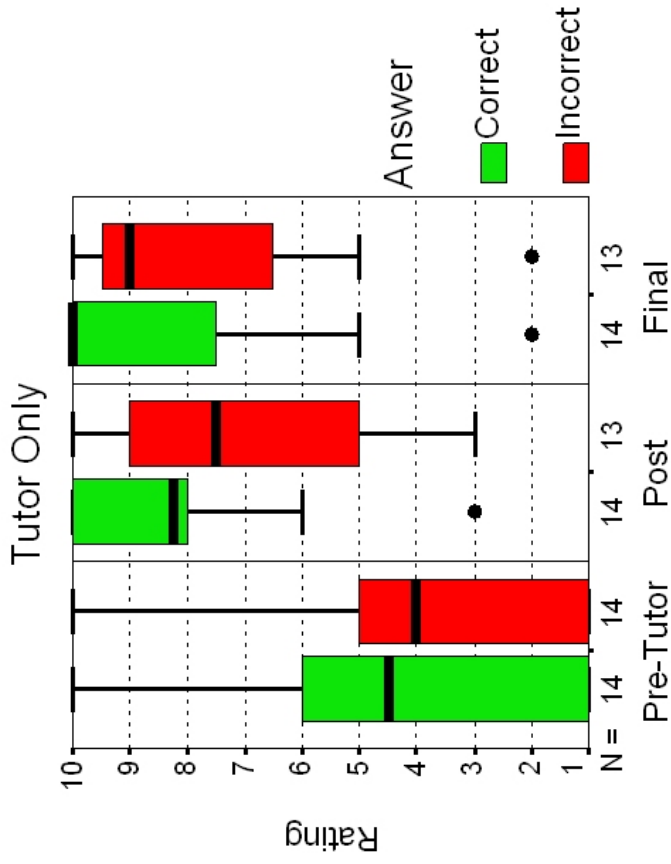
— Pre-Tutor ■ Post-Tutor ▲ Lecture+Interteaching



1 2 3 4 5 6 7 8 9 10 11 12 13 14

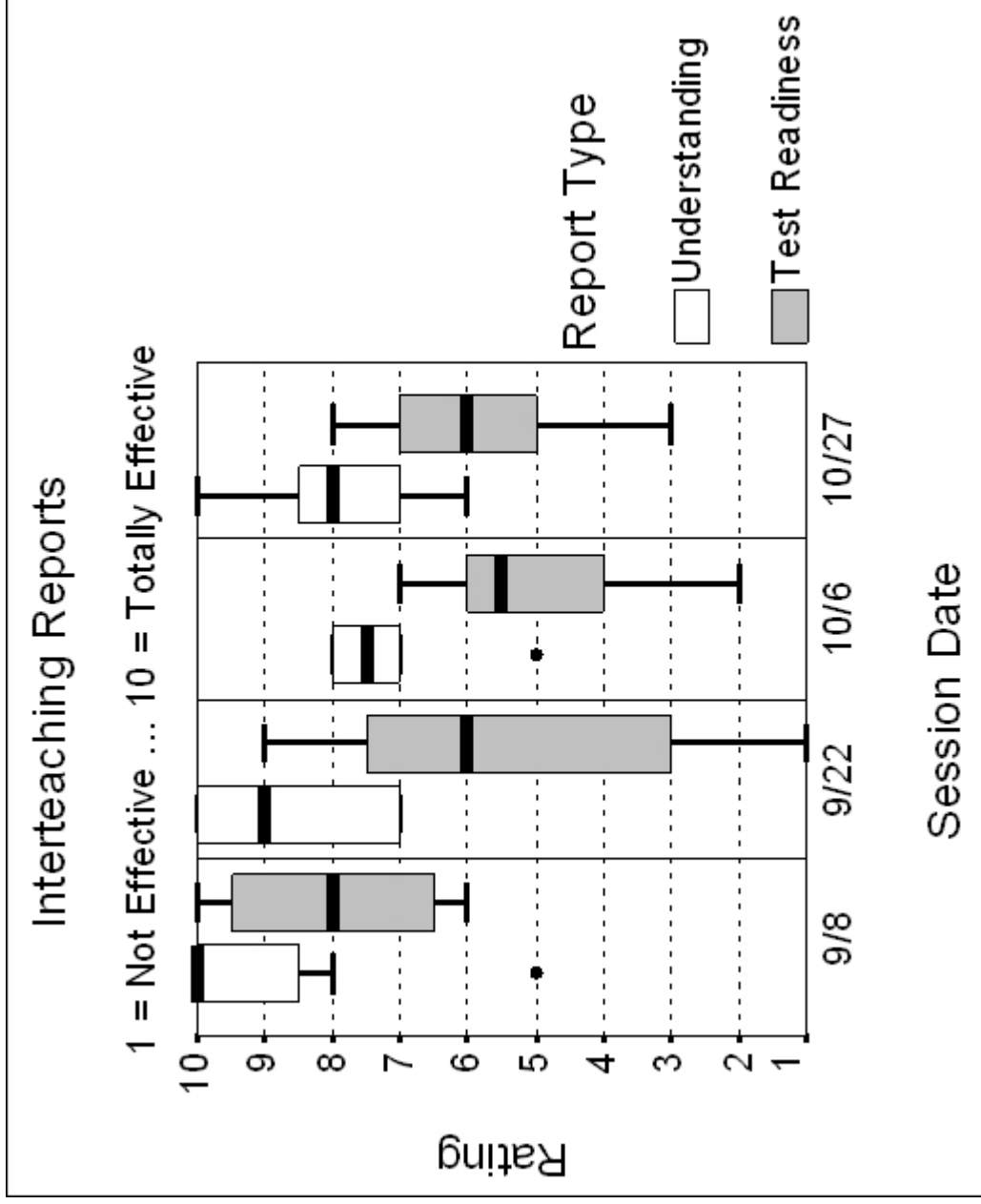
Tutor + Interteaching Class Members

Confidence in Rules Test Answers

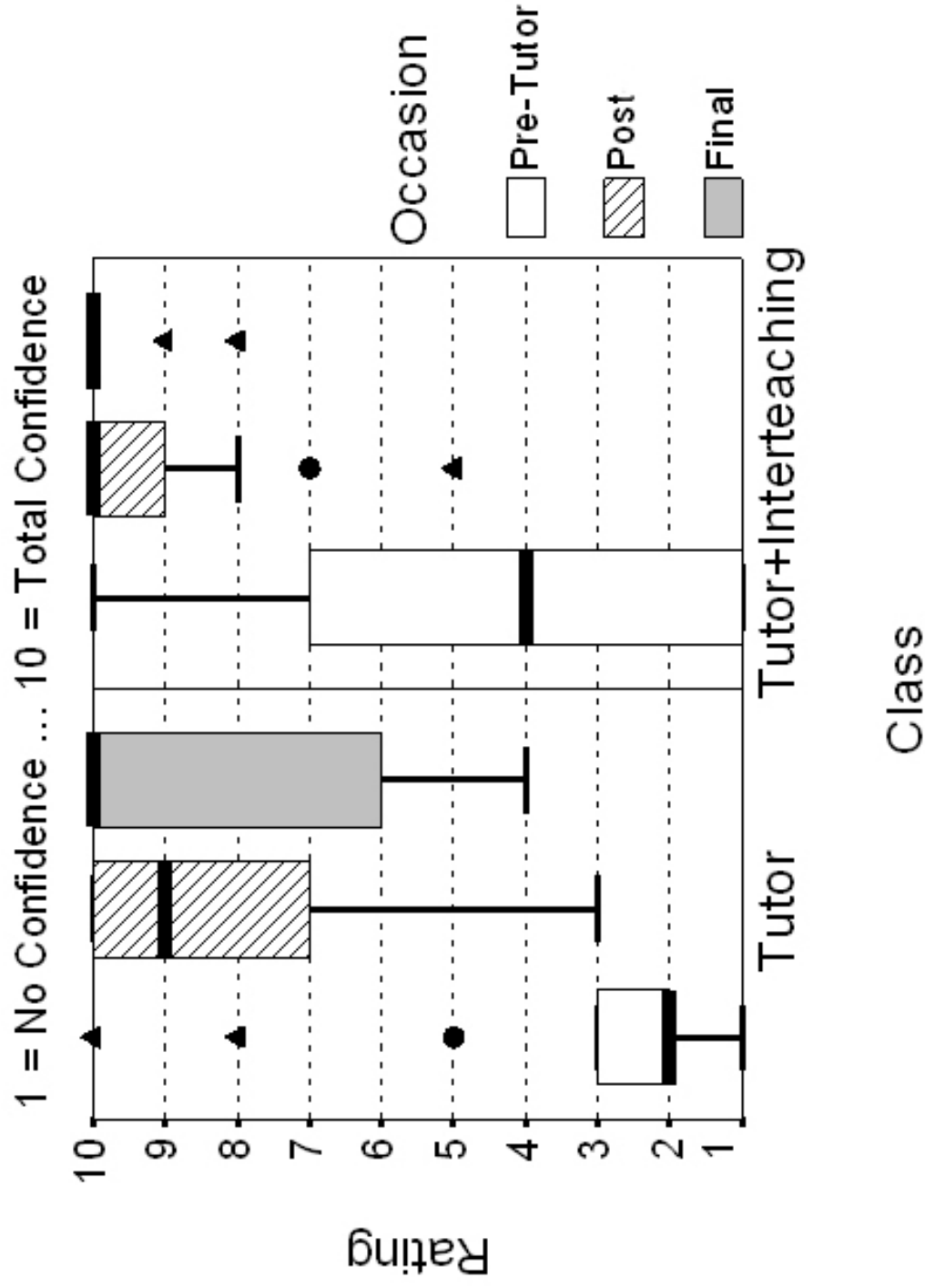


1 = No confidence ... 10 = Total confidence

Interteaching Reports



Software Self-Efficacy



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.

Challenges with Interteaching

- A rare student will show an aversion to collaborative learning.
- Pairs of students need different amounts of time.
- It is difficult to assess the “quality” of a collaboration objectively.

Challenges with Lecture

- I have to know what I'm talking about.

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?


The tutor, the source code, and all instructional material are freely available on the web.

EMURIAN - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://userpages.umbc.edu/~emurian/

Go



[Course Information](#) [My Schedule](#) [Henry](#) [UMBC](#)

[UMBC Java Tutoring System](#)

[Send an email to:
henry h. emurian](#)

[Association for Behavior Analysis](#)
[American Psychological Association](#)
[Information Resources Management Association](#)