

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

Teaching Java™: Managing Instructional Tactics to Optimize Student Learning



Henry H. Emurian
Information Systems Department
College of Engineering and
Information Technology

UMBC

1000 Hilltop Circle
Baltimore, Maryland 21250

In the News

New York Times: May 10, 2007
Harvard Task Force Calls for New Focus on Teaching and Not Just Research
By [SARA RIMER](#)

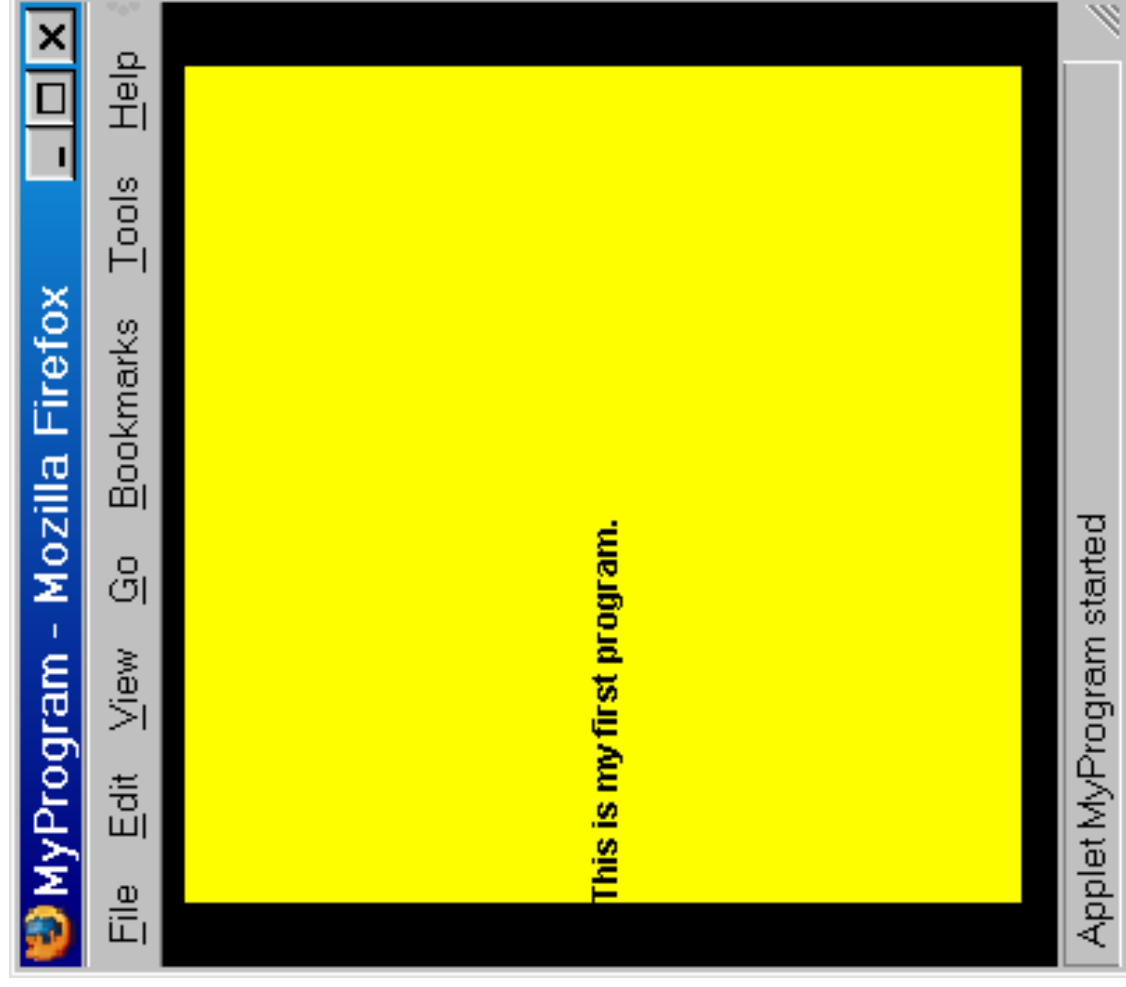
“ Direct instruction techniques [are] the most effective along all measures.”
Kim & Axelrod (2007)

Kim, T., & Axelrod, S. (2007). Direct instruction: an educators’ guide and a plea for action. *Behavior Analysis Review*, 2, 83-97.

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

```
1. import javax.swing.JApplet;  
2. import javax.swing.JLabel;  
3. import java.awt.Color;  
4. public class MyProgram extends JApplet {  
5.     JLabel myLabel;  
6.     public void init() {  
7.         myLabel=new JLabel("This is my first program.");  
8.         getContentPane().setBackground(Color.YELLOW);  
9.         getContentPane().add(myLabel);  
10.    }  
11. }
```

- Near transfer (understand and recite)
- Far transfer (meaningful learning → solve novel problems)

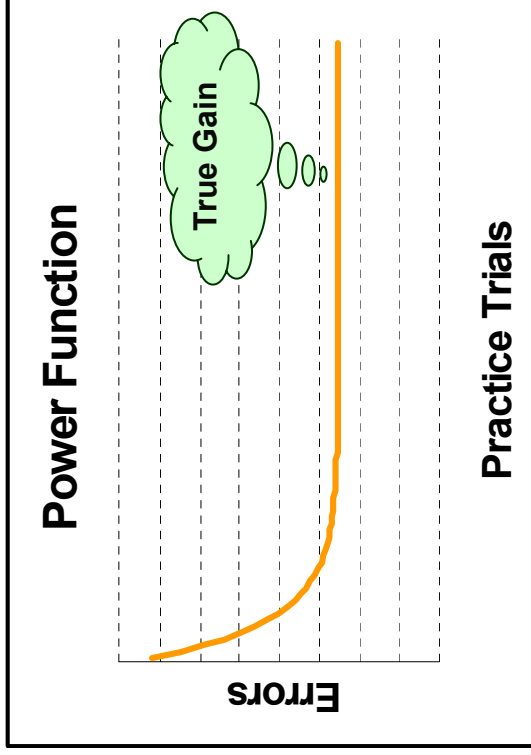


Challenges

- 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?

Model

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



Design-Based Research Methodology

- 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

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.

Lecture



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



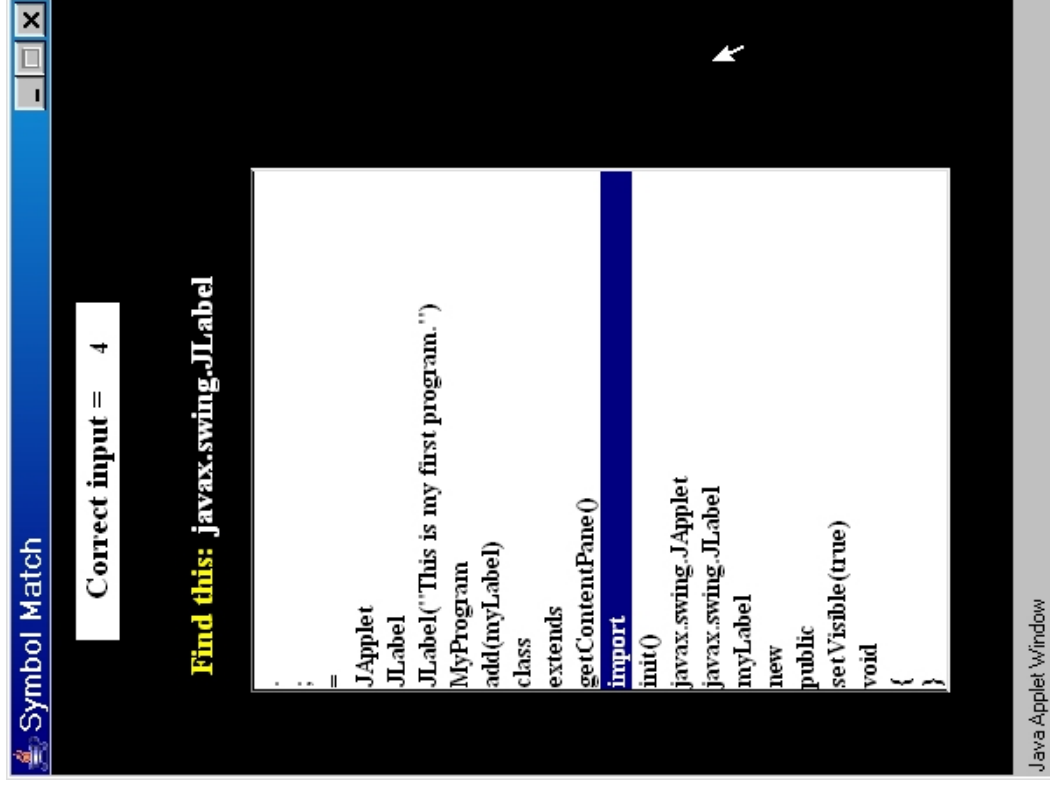
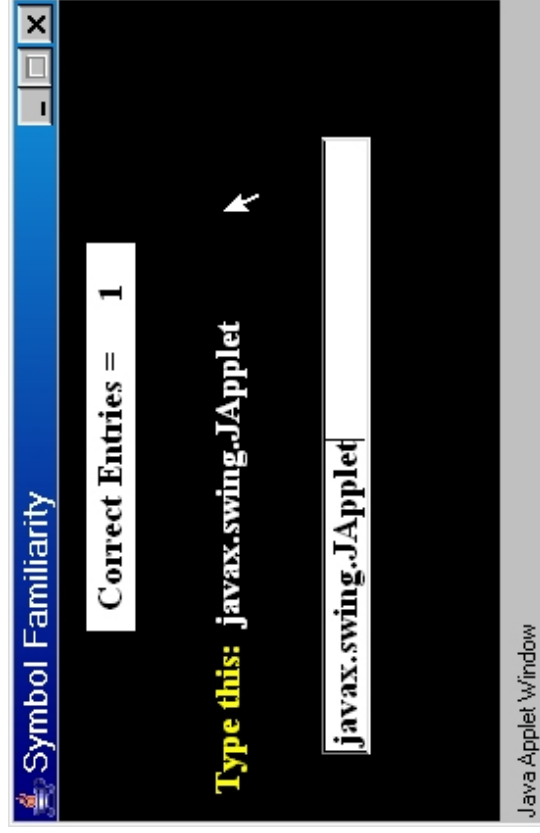
1. A **mutually probing, mutually informing conversation between two people** (Boyce & Hineline, 2002)
2. The questions on a topic to be addressed by the participants during a dialogue are prepared in advance by the teacher, and the **students come prepared to interteach**
3. Has the objective of insuring,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

Innovation

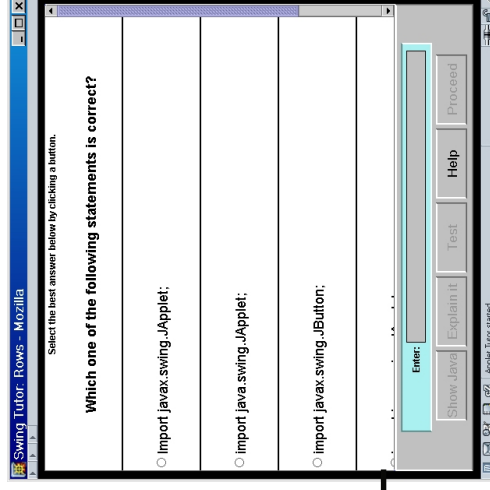
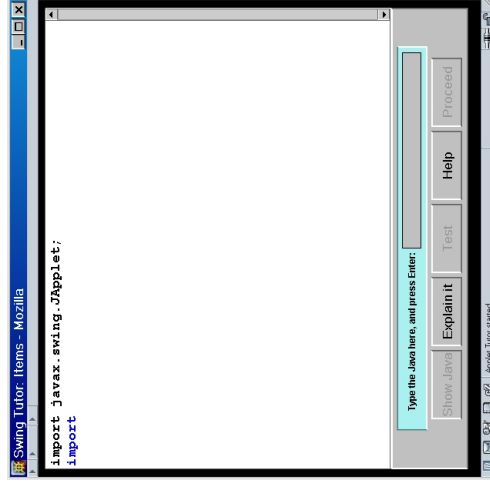
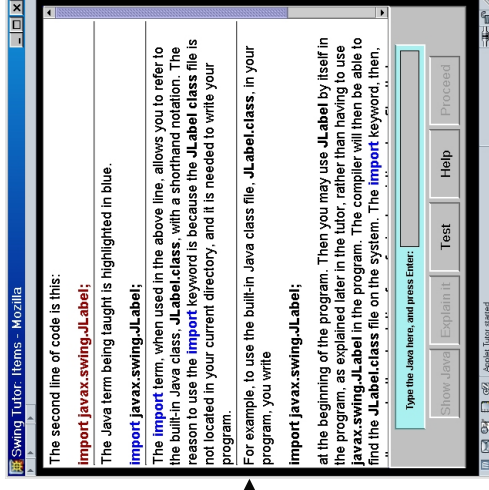
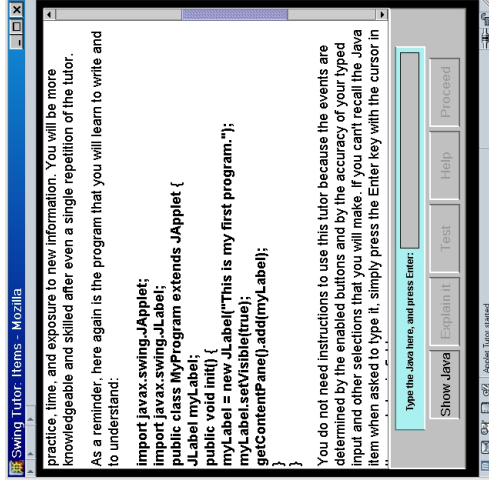
- In the current study, the innovation was to allow members of interteaching pairs to **collaborate on their answers** to the rule-based questions. Answers did not have to match, however.
- On the day following the interteaching session, the **instructor posted the correct answers** to the rule-based questions along with an explanation of the principle involved.

Programmed Instruction Tutoring System

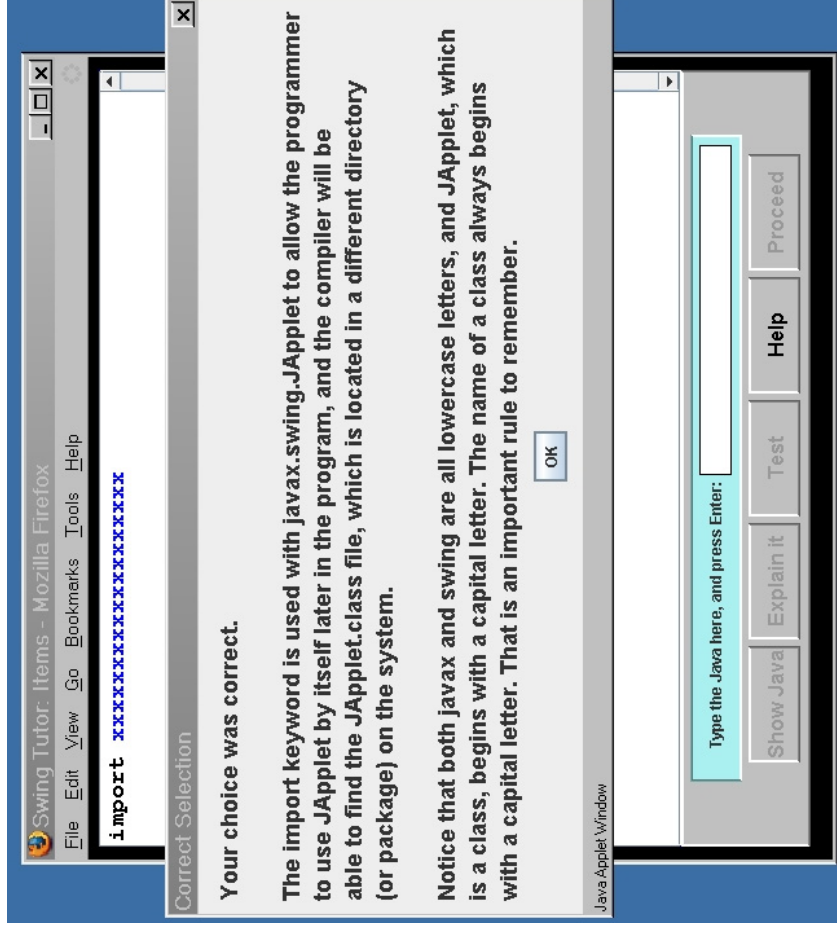
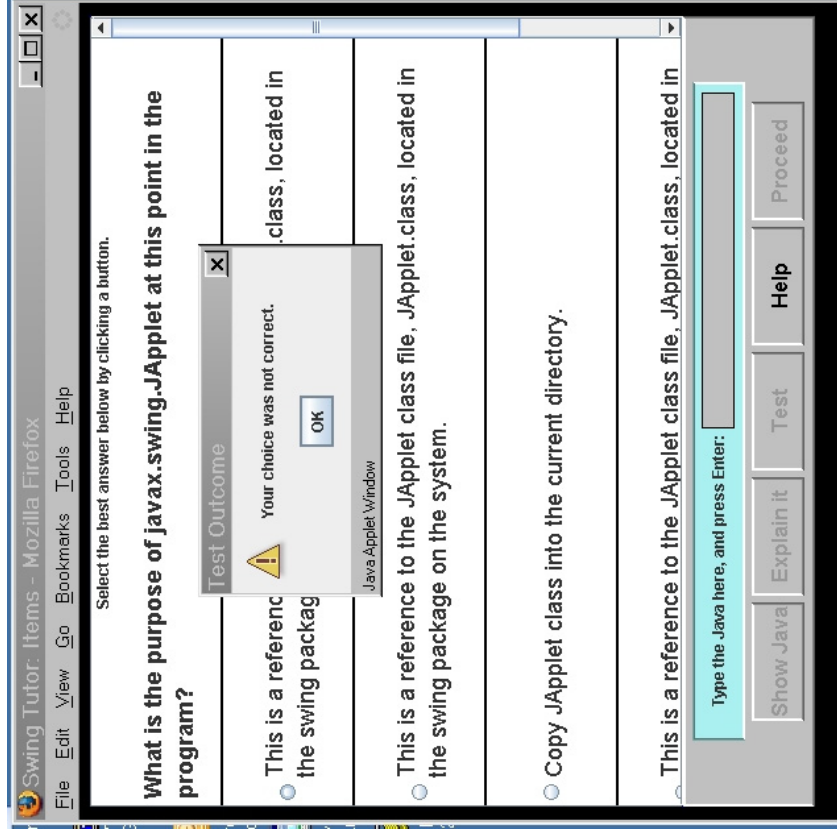
Basics



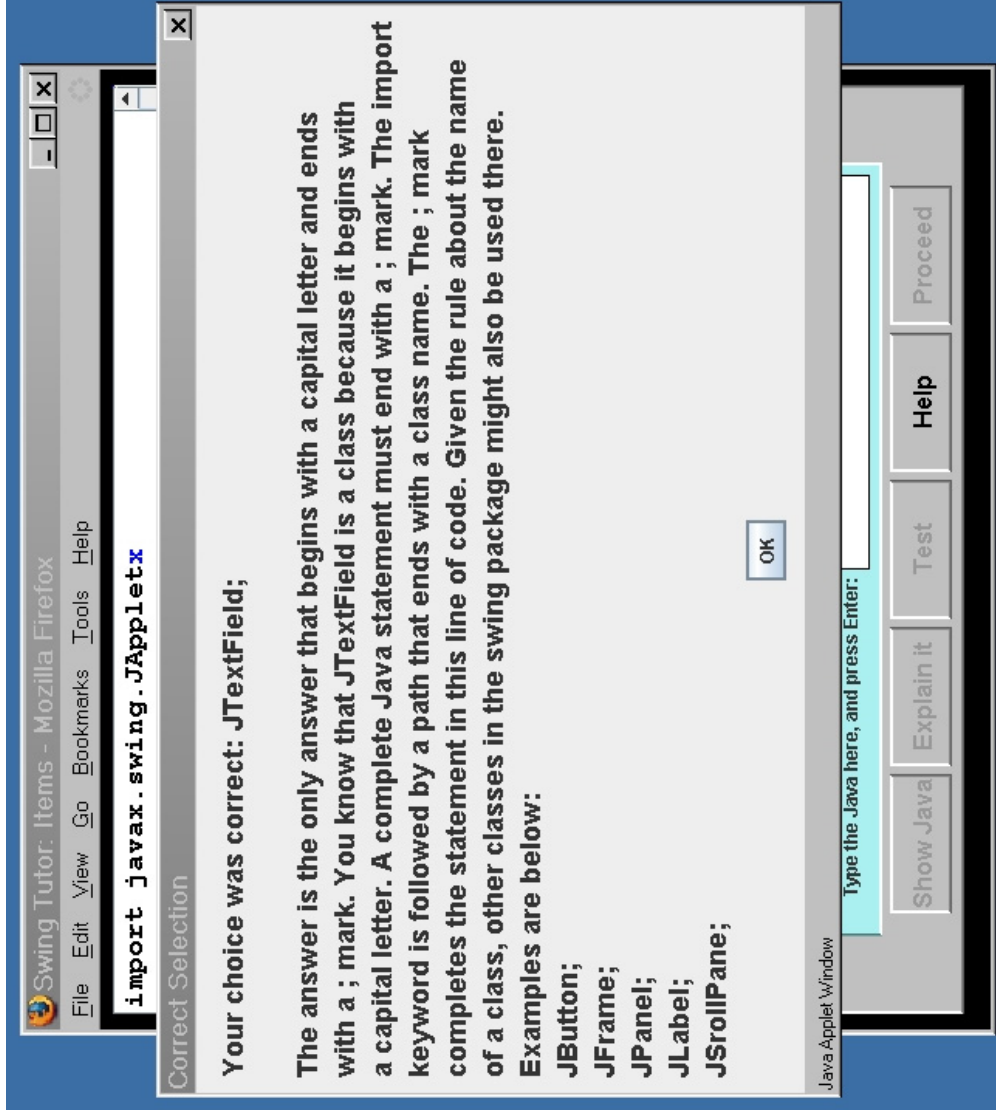
Java Tutor: Item Learning

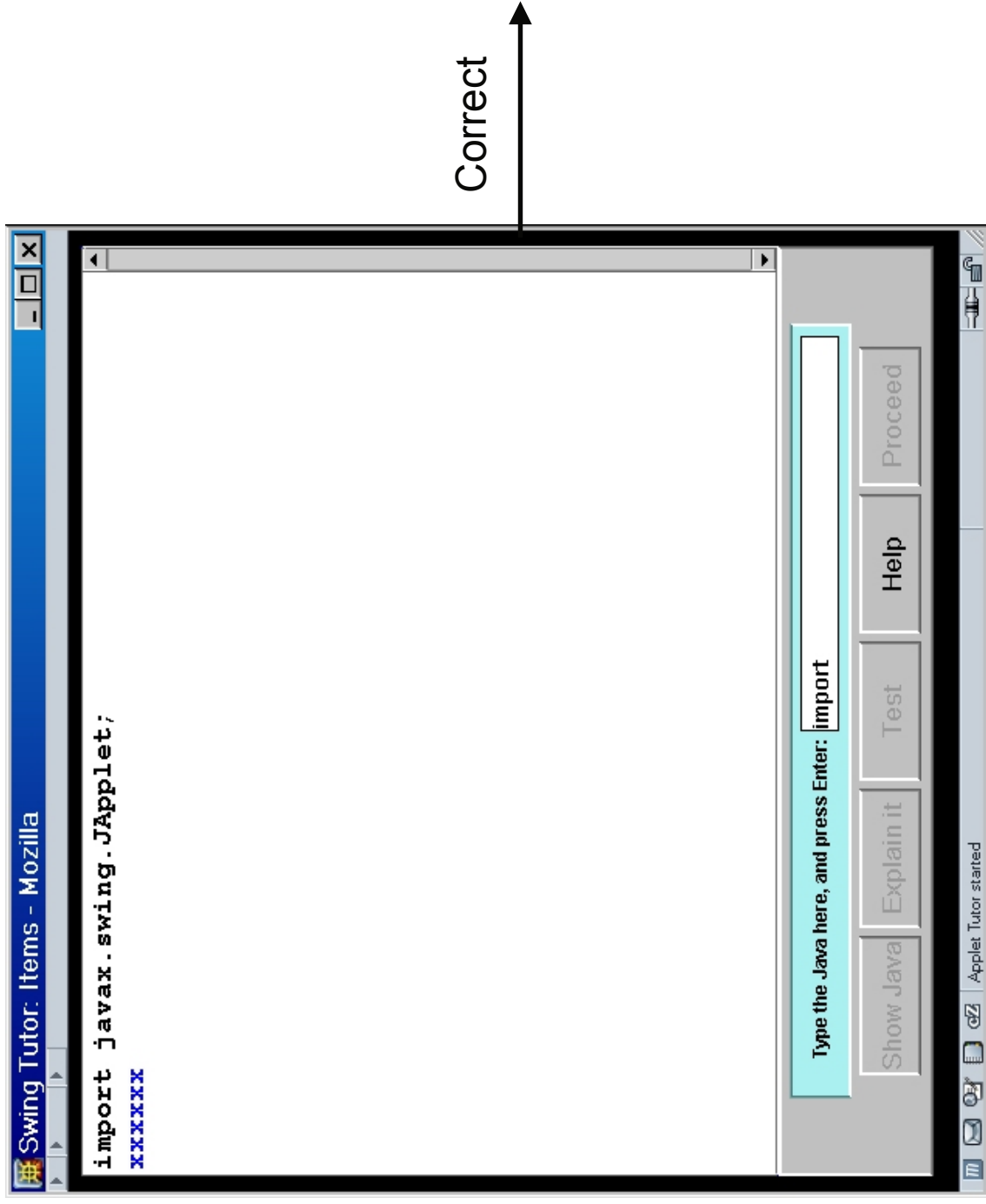


Item Test Outcomes



Multiple Exemplars

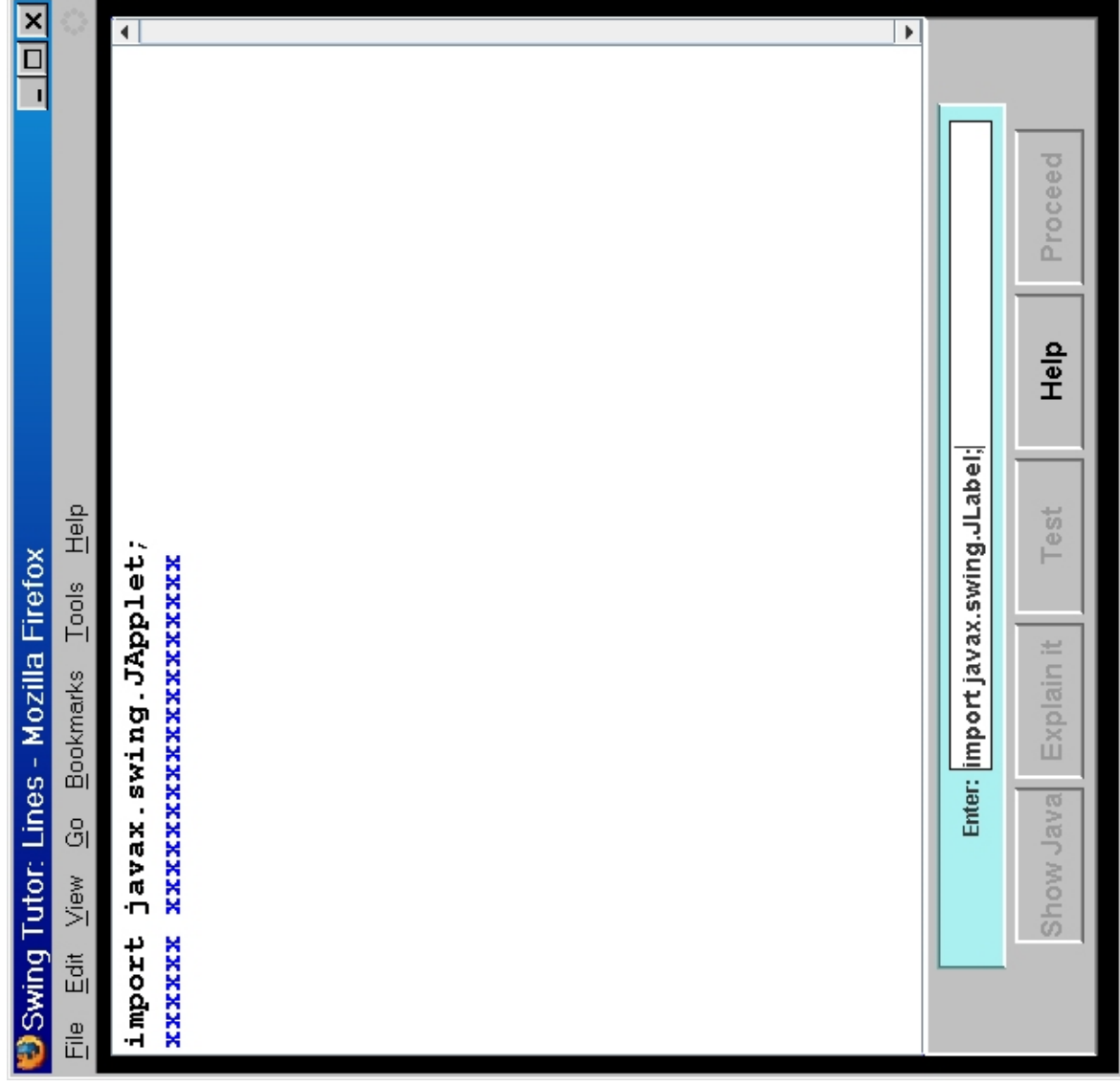




Incorrect:
View
Correct
Input

Incorrect Input Feedback





Type the program in the white space below, and select the Submit button. Do not use the Tab key. Please do not use notes. Try to enter the program from your memory. If you can't remember the program, just select Submit, and you can see the code again.

```
import javax.swing.JApplet;  
import javax.swing.JLabel;  
import java.awt.Color;  
public class MyProgram extends JApplet {  
    JLabel myLabel;  
    public void init() {  
        myLabel = new JLabel("This is my first program.");  
        getContentPane().setBackground(Color.YELLOW);  
        getContentPane().add(myLabel);  
    }  
}
```

Java Applet Window

Enter the Java Program

Type the program in the white space below, and select the Submit button. Do not use the Tab key. Please do not use notes. Try to enter the program from your memory. If you can't remember the program, just select Submit, and you can see the code again.

```
import javax.swing.JApplet;
import javax.swing.JLabel;
import java.awt.Color;

public class MyProgram extends Applet {
    JLabel myLabel;

    public void init() {
        myLabel = new JLabel("This is my first program.");
        getContentPane().setBackground(Color.YELLOW);
        getContentPane().add(myLabel);
    }
}
```

Submit

Clear

Java Applet Window

Error

Incorrect Input

Your input was not correct. Examine the below code and see if you can spot your error. The below code has the same format as the previous tutor code, but you do not have to use that format in the adjacent window. Select the Noted button when you are ready to try again.

```
import javax.swing.JApplet;
import javax.swing.JLabel;
import java.awt.Color;

public class MyProgram extends JApplet {
    JLabel myLabel;

    public void init() {
        myLabel = new JLabel("This is my first program.");
        getContentPane().setBackground(Color.YELLOW);
        getContentPane().add(myLabel);
    }
}
```

Noted

Java Applet Window

Interteaching Report

IS 413

Interteaching Report #1

Your name xxxx Date xxxxx

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 javax.swing.JApplet;  
import javax.swing.JLabel;  
import java.awt.Color;  
public class MyProgram extends JApplet {  
    JLabel myLabel;  
    public void init() {  
        myLabel = new JLabel("This is my first program.");  
        getContentPane().setBackground(Color.YELLOW);  
        getContentPane().add(myLabel);  
    }  
}
```

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:

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 questions correctly.

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

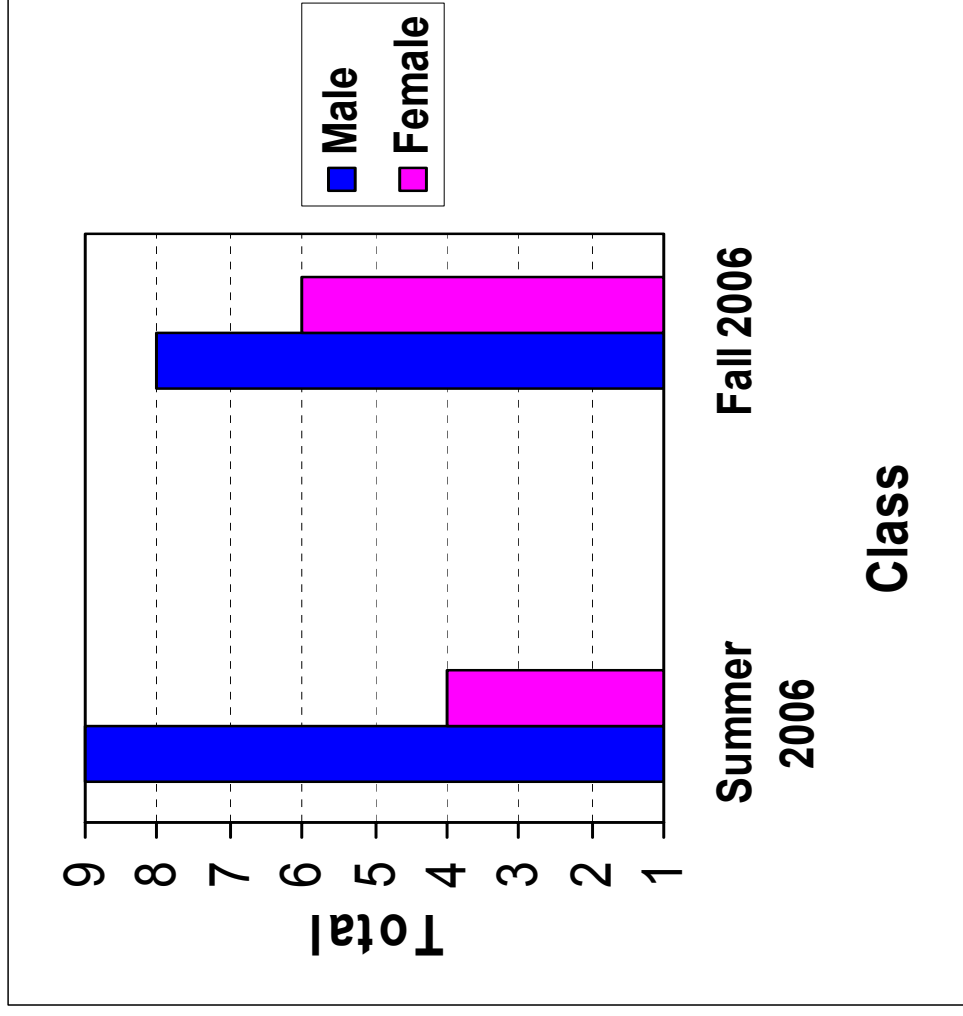
(Not confident) 1 2 3 4 5 6 7 8 9 10 (Totally confident)

Enter one number that describes your confidence:

Procedure

Class 1	–Individual Questionnaires Tutor –Individual Questionnaires
Class 2	Lecture –Run the applet –Individual Questionnaires
Class 3	Interteaching –Collaborative Questionnaires –Posting answers to rule-based questions
Class 4	Quiz –48 Questions from Java tutor –12 Rule-based questions from questionnaires

Students



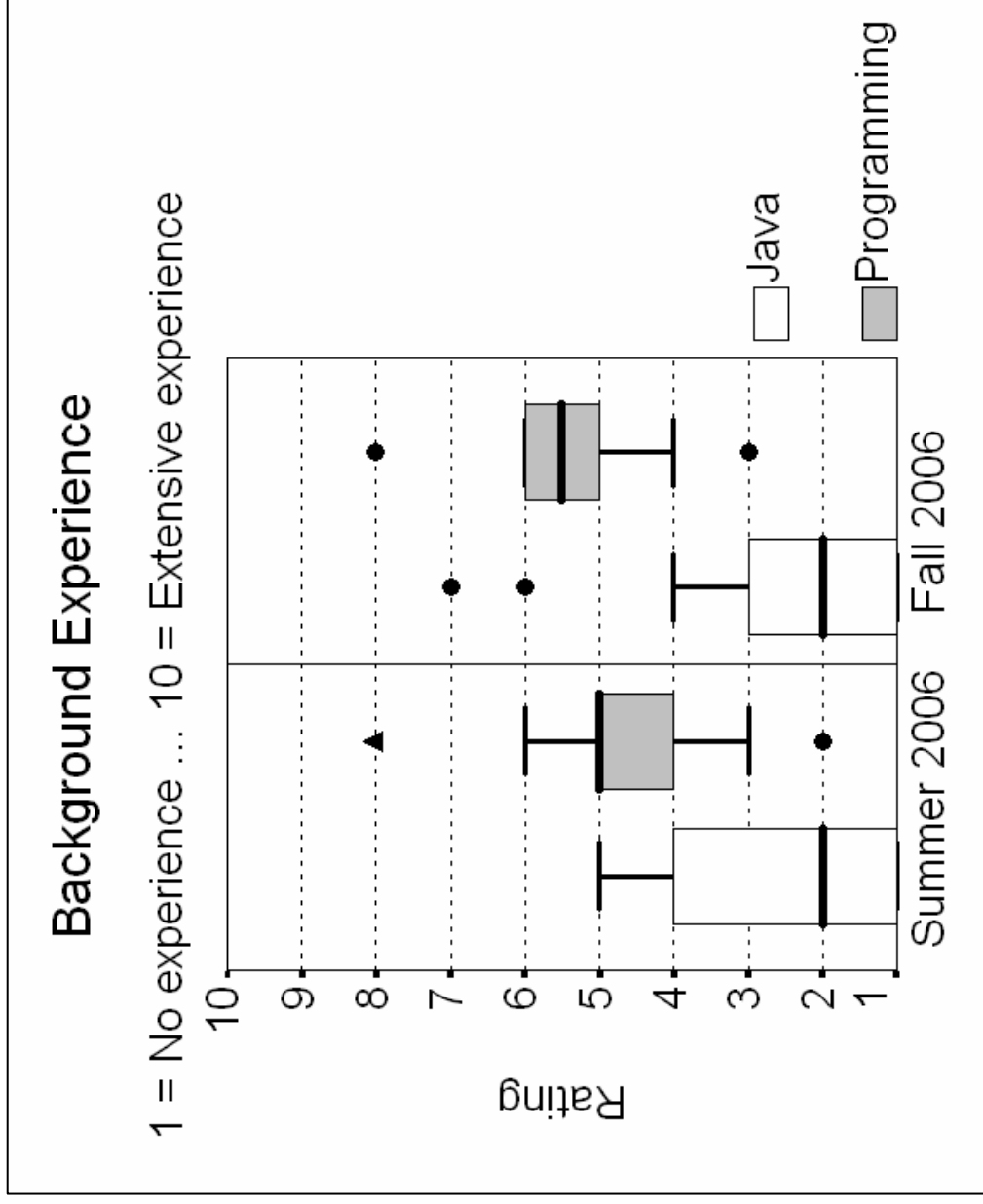
Summer 2006

Median = 28 yrs,
Range = 23 to 33

Fall 2006

Median = 22 yrs,
Range = 21 to 32

Students



1. Which of the following lines most likely would be used to create a shorthand notation for the compiler to locate the JFrame class, which is built-in to Java?

a. import ../class/JFrame;
b. access JFrame.class;
c. import javax.swing.JFrame;
d. import java.awt.JFrame.class;
e. append javax.swing.JFrame;

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:

2. Which of the following lines most likely would be used to construct an instance of the JButton class?

a. JButton = new JButton("Hello");
b. myButton = new JButton("Click Me");
c. Button = new JButton("Hello");
d. myButton = JButton.class("Hello");
e. myButton = new JButton("Click Me").

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:

Software Self-Efficacy Ratings

Question 9

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

getContentPane()

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

Enter a number here:

Question 10

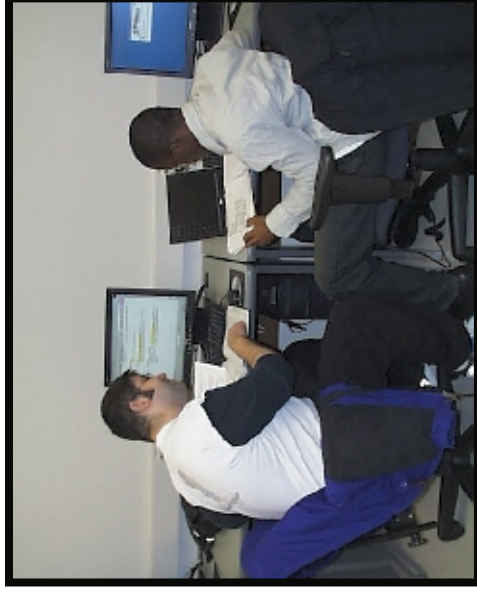
How confident are you that you understand and can use the following symbol now to write a Java program?

import

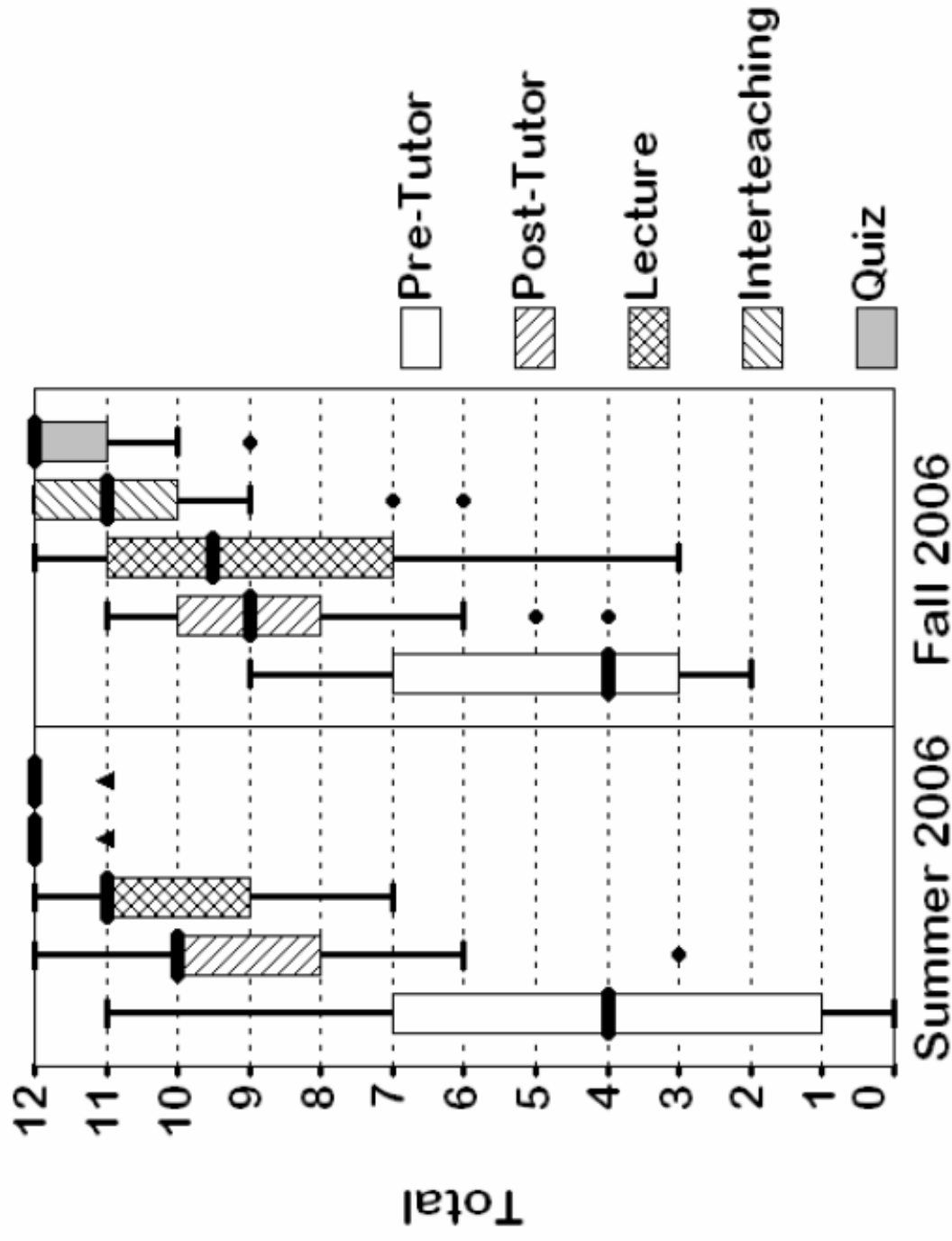
Not at all confident. 1 2 3 4 5 6 7 8 9 10 Totally confident.

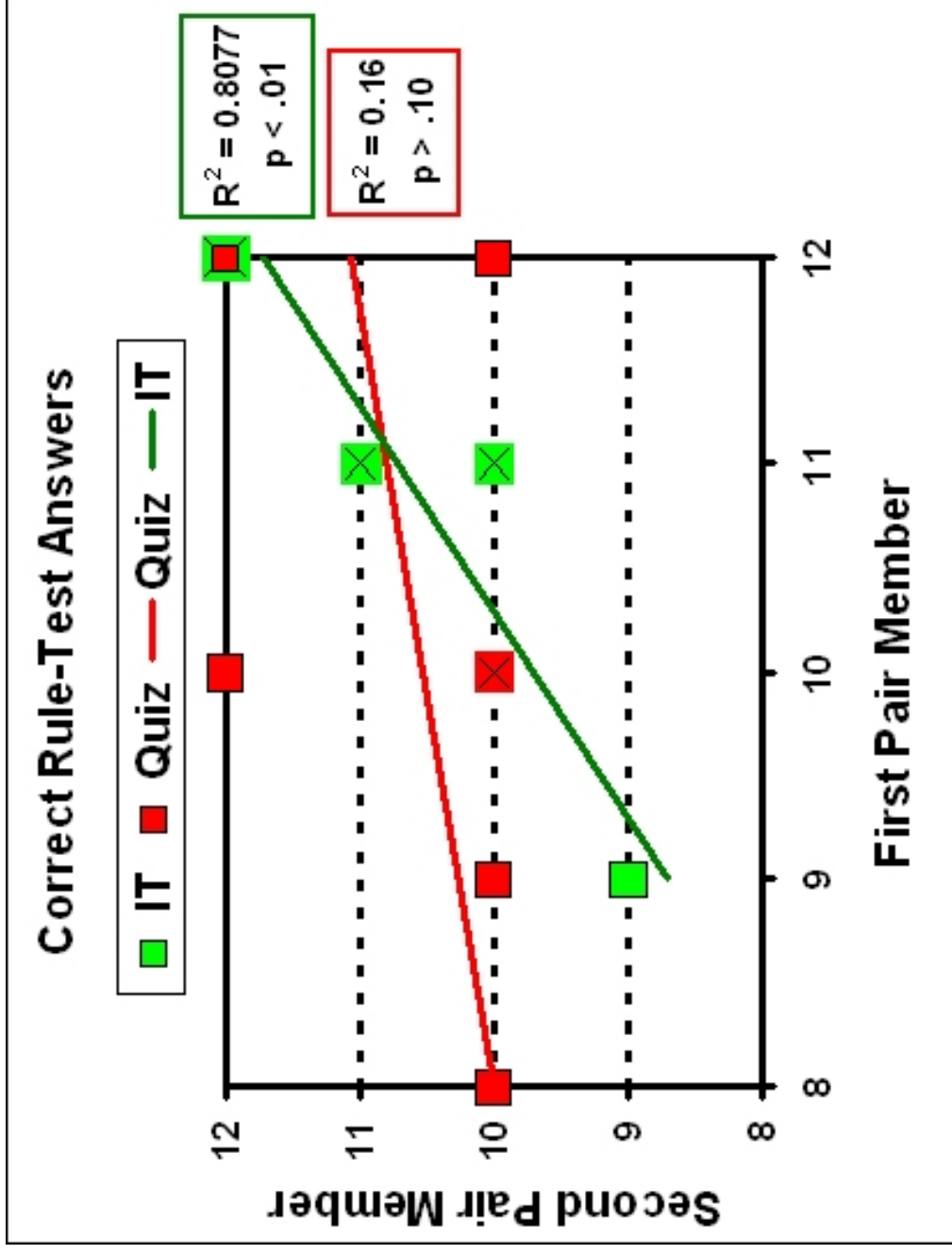
Enter a number here:

Interteachers in Action



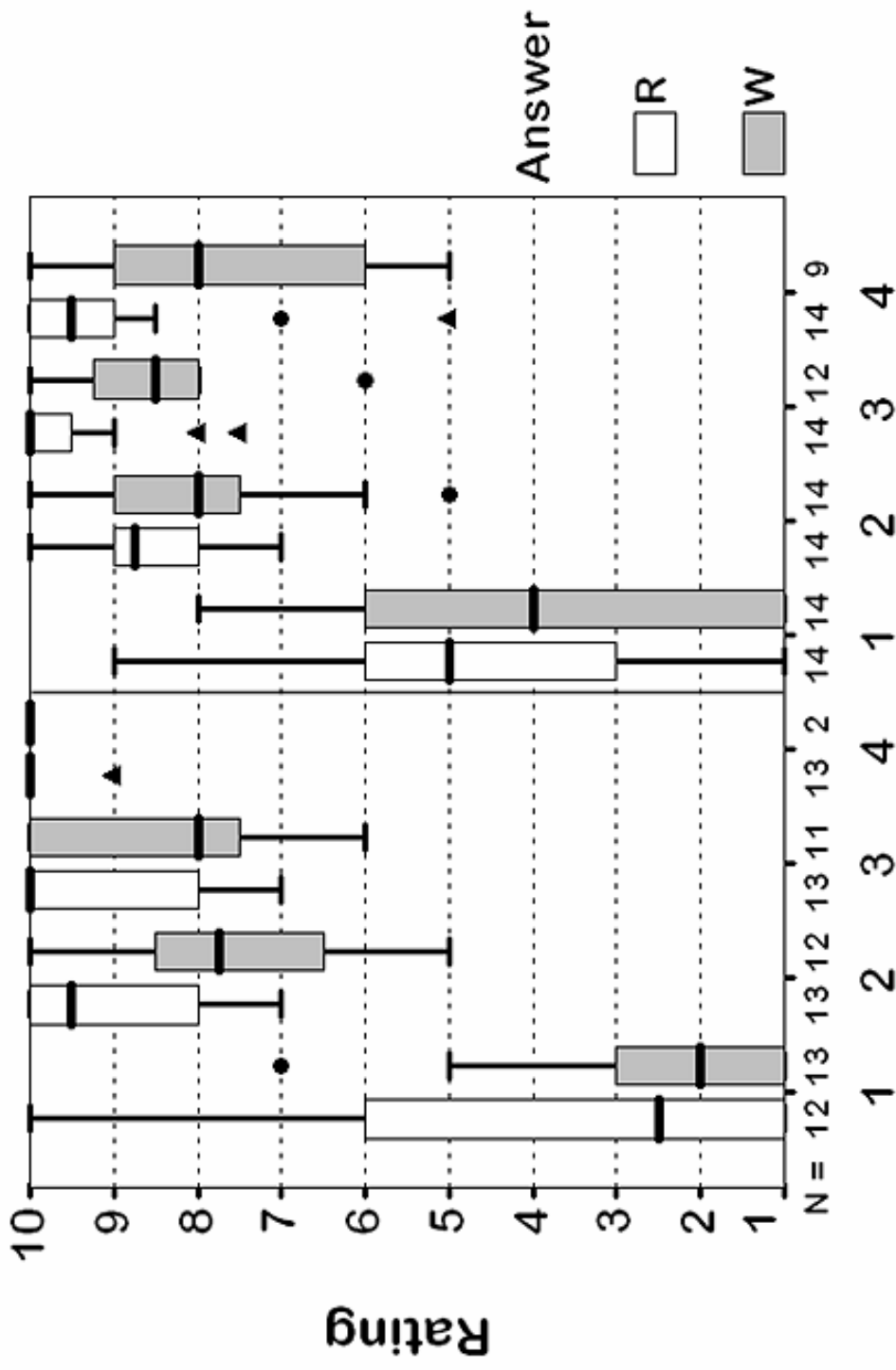
Correct Answers on Rule Test



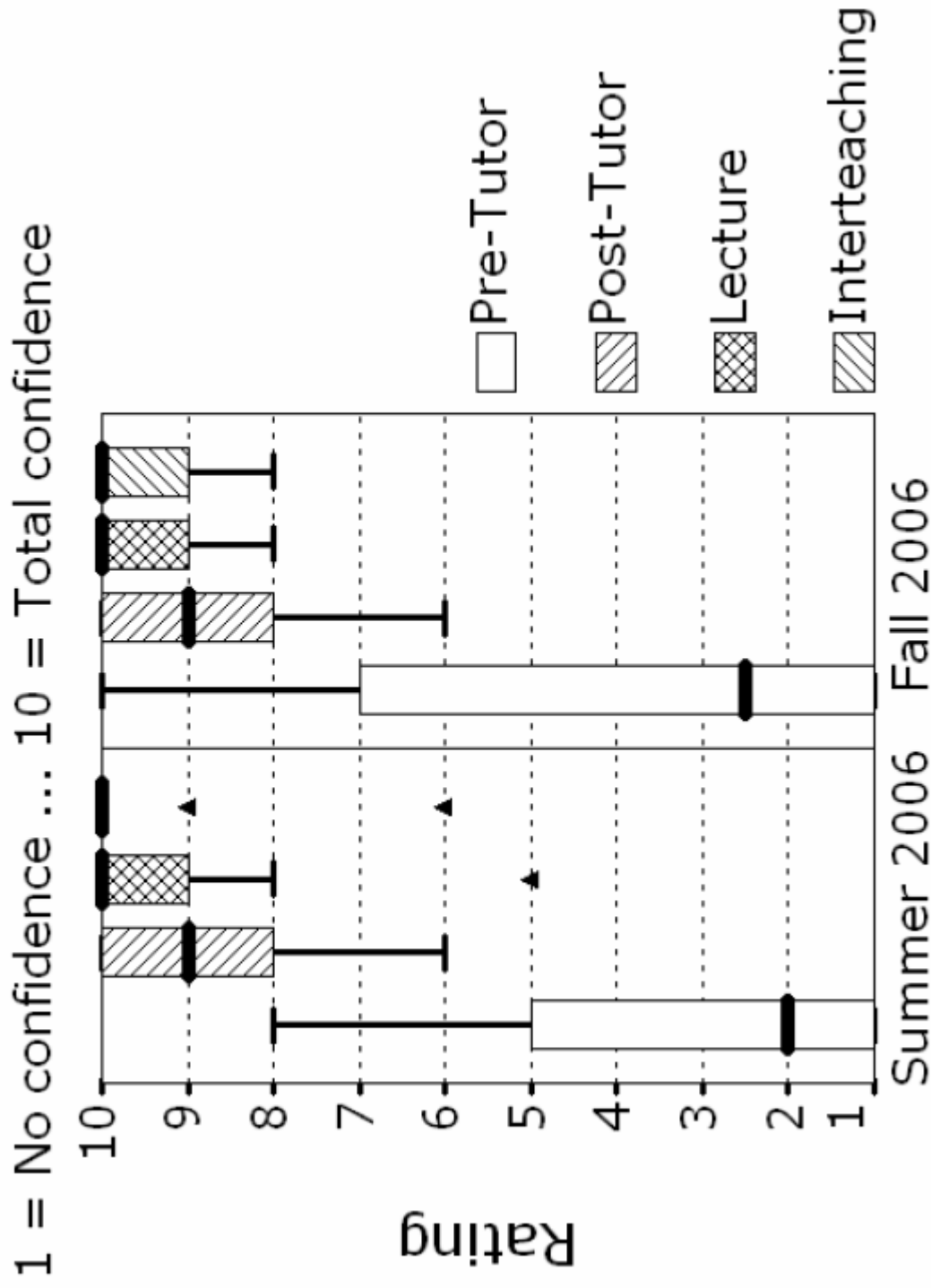


Confidence in Rule Answers

Summer 2006 Fall 2006

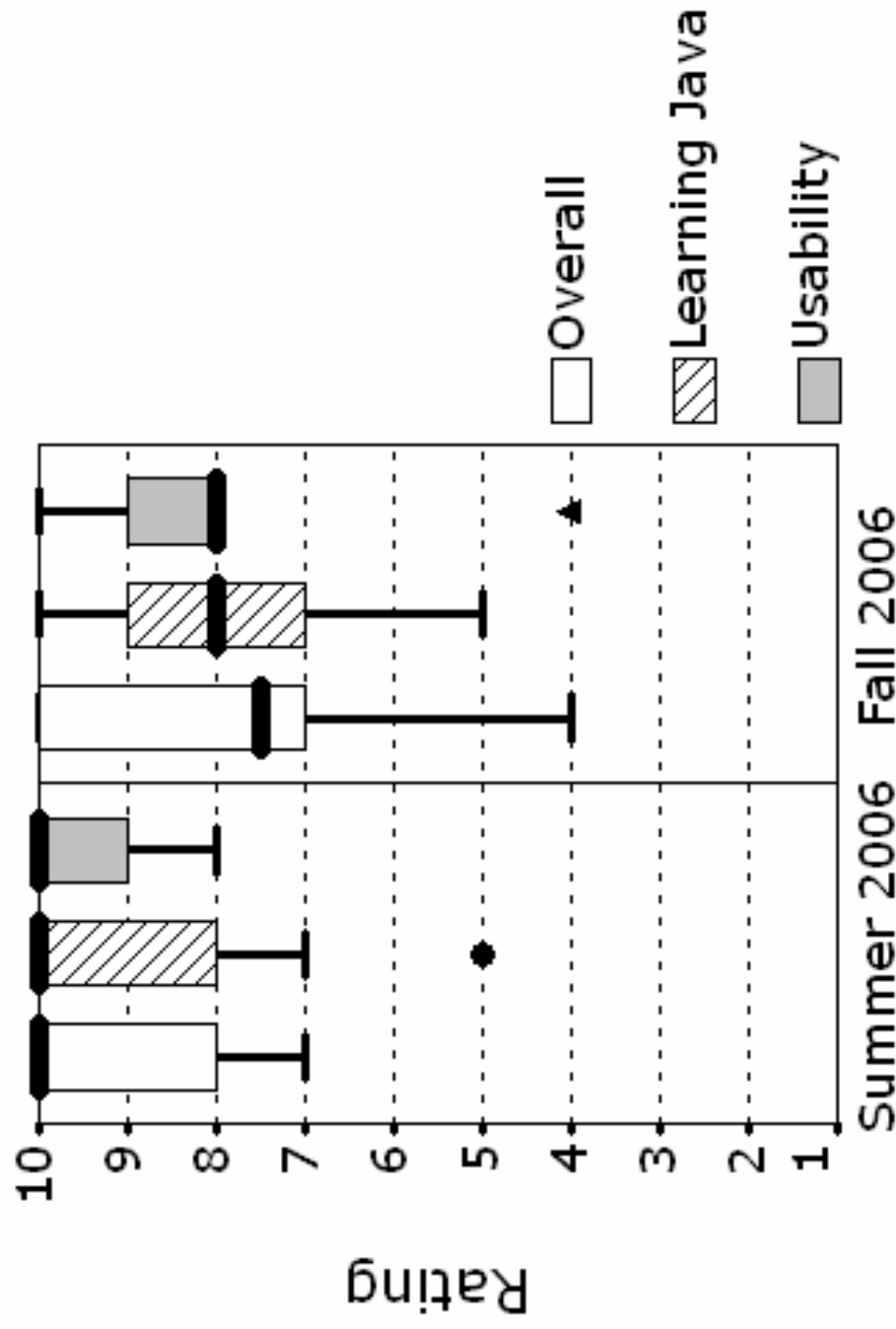


Software Self-Efficacy

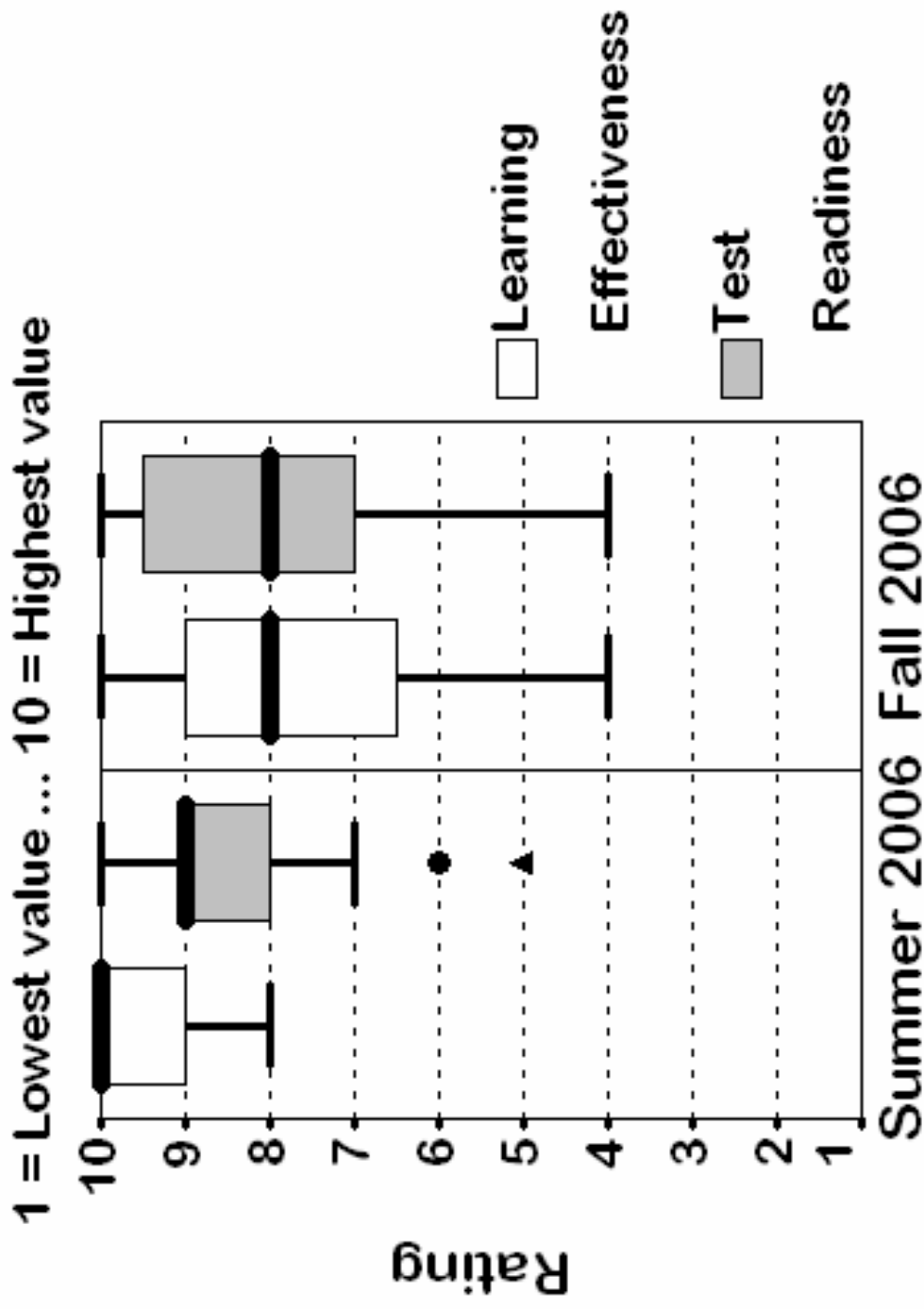


Tutor Evaluation

1 = Totally negative ... 10 = Totally positive



Interteaching Evaluation



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, until a multiple-choice item is answered correctly, 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, but there are issues of retention.
3. The competency attained sets the occasion for advanced learning with enthusiasm.
4. Students like the tutor and the interteaching, and so do I.
5. I also like to use lectures with hands-on learning and classroom collaboration among students.

Thank you!

Questions?

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

<http://nasa1.ifsm.umbc.edu/learnJava/tutorLinks/TutorLinks.html>