

Performance on a Web-Based Tutoring System for Java Training

Henry H. Emurian
Information Systems
UMBC
1000 Hilltop Circle
Baltimore, Maryland 21250

Ashley G. Durham
Centers for Medicare and
Medicaid Services
7500 Security Boulevard
Baltimore, Maryland 21244

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. }
```

Challenges

- ***Student Motivation***
 - Information Systems students are less keen to study computer programming than are computer science students.
- ***Student Preparation***
 - Information Systems students are less prepared to study computer programming than are computer science students.

Our Values

- **Rote memorization** is fundamental to the acquisition of skills that set the occasion for “understanding.”
 - Constructivism comes later.
- **Disciplined study behavior** is essential to acquire skill, and many, if not most students, do not know how to study.
- **Repetition** and overlearning are essential to the learning process.
- **Feeling good about yourself** after hard work sustains enthusiasm for learning.

“Regurgitation” and Beyond...

- ***Regurgitation is Good***
 - Mindful, intentional, informed, purposeful, and accurate regurgitation.
- ***General Rules***
 - Within the context of the interactive tutor, **frames of information** about general syntax and semantics are presented.
 - **Multiple-choice tests** based on these frames are embedded throughout the tutor.

How it all started (circa 1987)

1. program circle
2. real r, area
3. c This program reads a real number r and prints
4. c the area of a circle with radius r.
5. write (*,*) 'Enter the radius r = '
6. read (*,*) r
7. area = 3.14159*r*r
8. write (*,*) 'Area = ', area
9. stop
10. end

What I used to do:

- **Lecture**
 - Write the code on the blackboard and explain it.
- **Exhort**
 - Tell the students to learn a program for a test.
- **Observe**
 - Individual differences in test performance.
- **Neglect**
 - I did not teach students how to learn.

Enlightened Approach

```
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. }
```

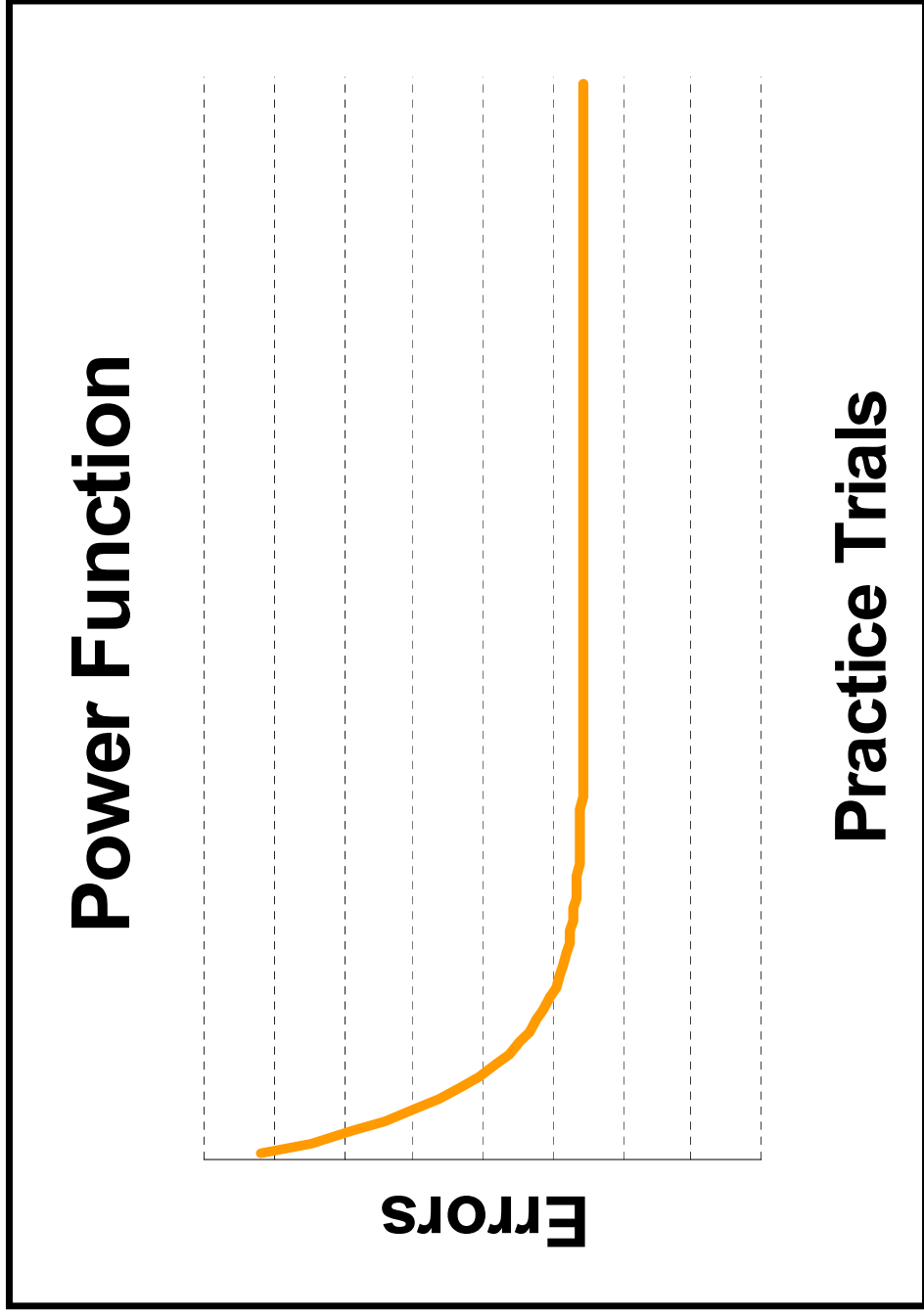
- Programmed instruction
- Personalized System of Instruction
- Successive approximations to mastery

Programmed Instruction

- A set of **structured interactions** between a learner and the material to be mastered.
- Structures study behavior that is focused on the **individual learner**.
- Manages the **moment-by-moment interactions** between a learner and a tutor.
- **Step-wise progression** from elementary knowledge units or facts to the achievement of a complex repertoire that is the objective of learning.

- Keller (1968): *Personalized System of Instruction*
 - Unit Perfection
 - Self-paced progression
 - Focus on written word to communicate information
 - Repeated testing of concepts
 - Collaboration with peers and experts

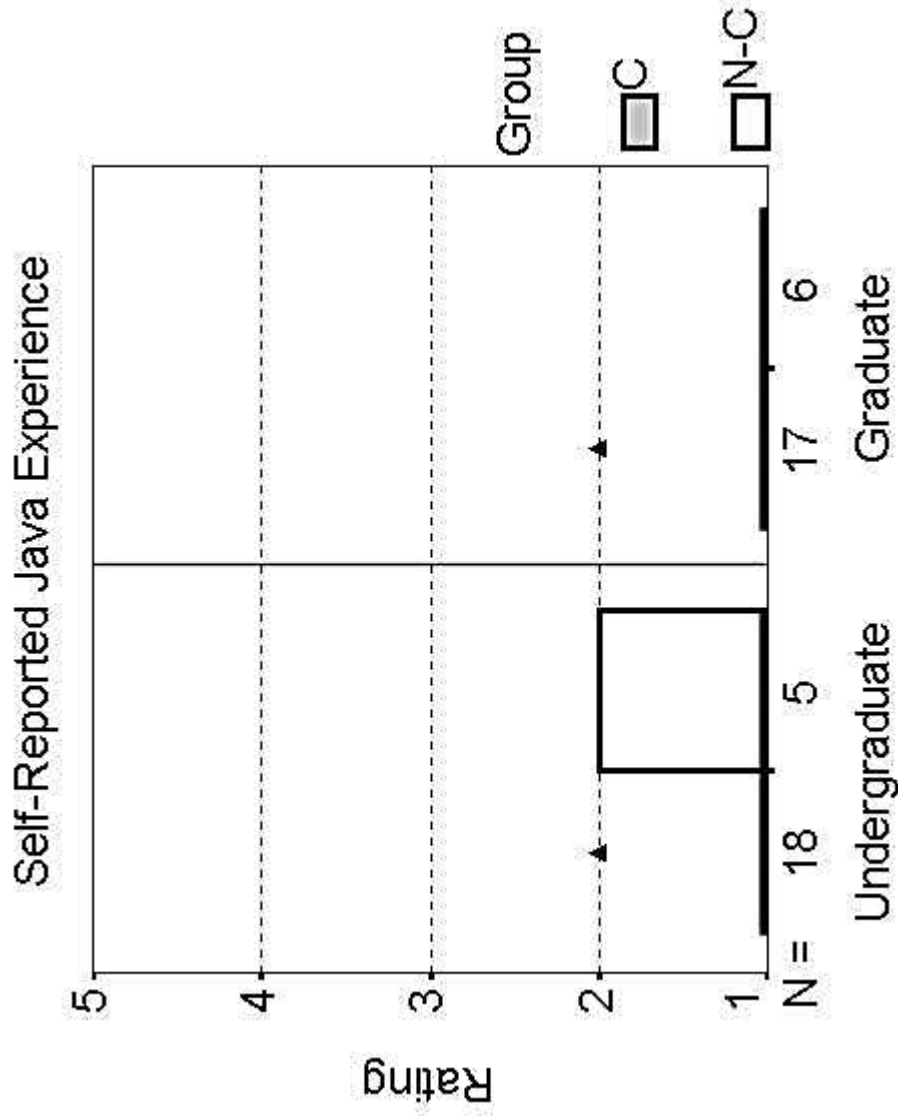
Acquisition Curve



Spring 2002 Course Sections

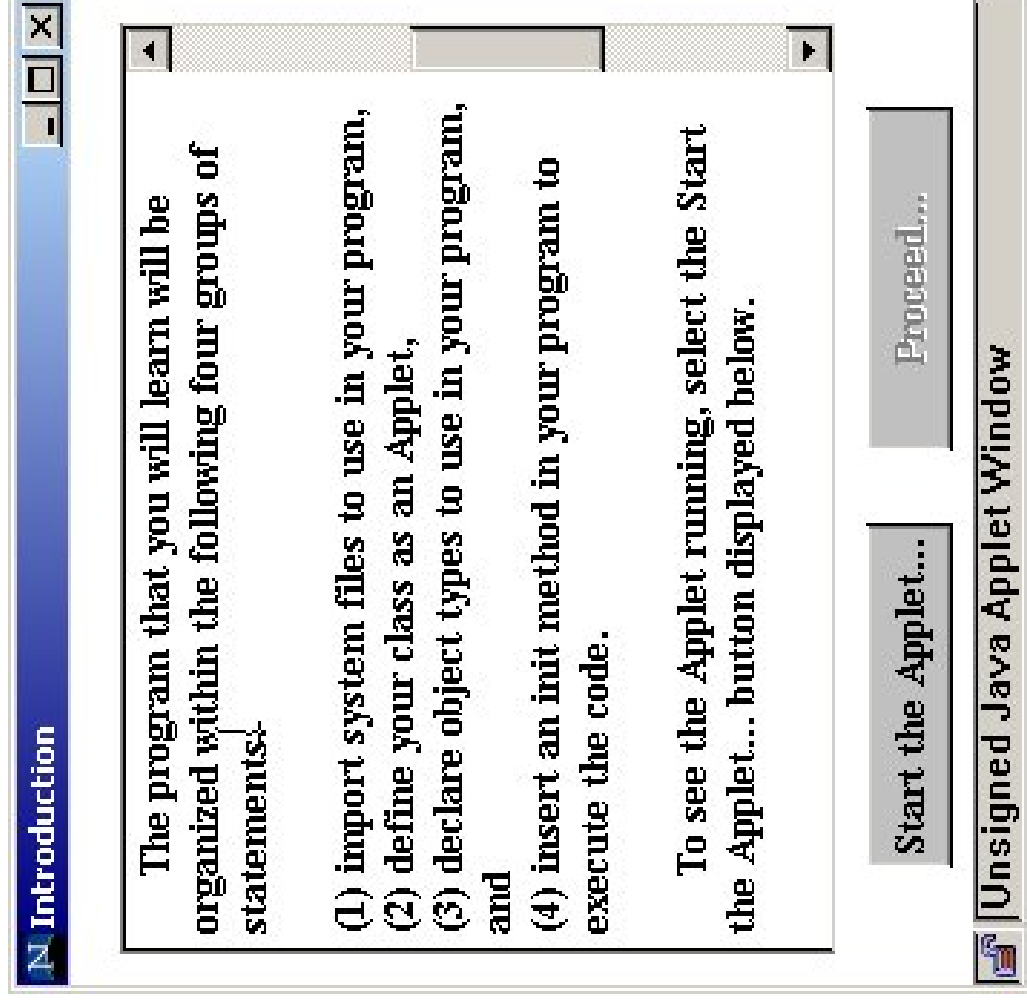
- **Undergraduate class** • **Class 1**
 - 13 F (median age = 22)
 - 10 M (median age = 22)
- **Graduate class**
 - 14 F (median age = 26)
 - 9 M (median age = 28)
- **Constraints**
 - Students rather than “subjects”
 - Fixed 2.5-hr class duration
 - 14 classes in the semester
 - Approach to the data
- **Class 2**
 - Run the Applet
- **Classes 3 – 14**
 - Lectures, demonstrations, supervision
- **Class 1**
 - Pre-tutor questionnaires
 - Java Experience
 - Confidence in Java
 - Run the tutor
 - Post-tutor questionnaires
 - Evaluate the tutor
 - Confidence in Java

Prior Experience with Java

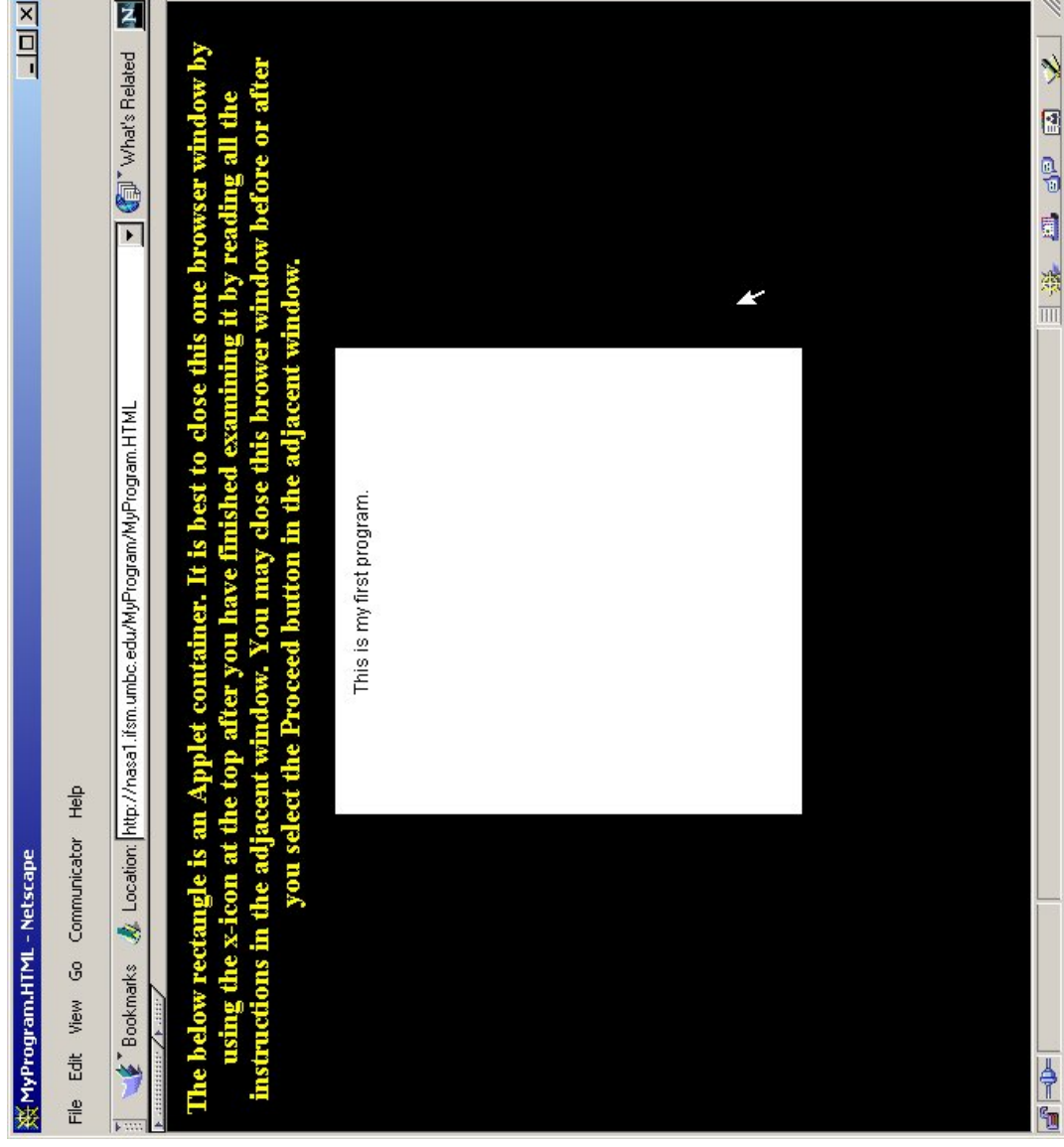


1 = No experience. (I am a novice in Java.)
5 = Extensive experience. (I am an expert in Java.)

Advanced Organizers



Run the Applet



Program Overview

The screenshot shows a window titled "Unsigned Java Applet Window". The window contains a text area with 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);
    }
}
```

Below the code editor is a text box containing the following text:

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.

Examine the adjacent lines of code to see the general appearance of a Java program and the types of symbols and expressions that appear. You are not expected to understand these lines of code

Proceed...

Unsigned Java Applet Window

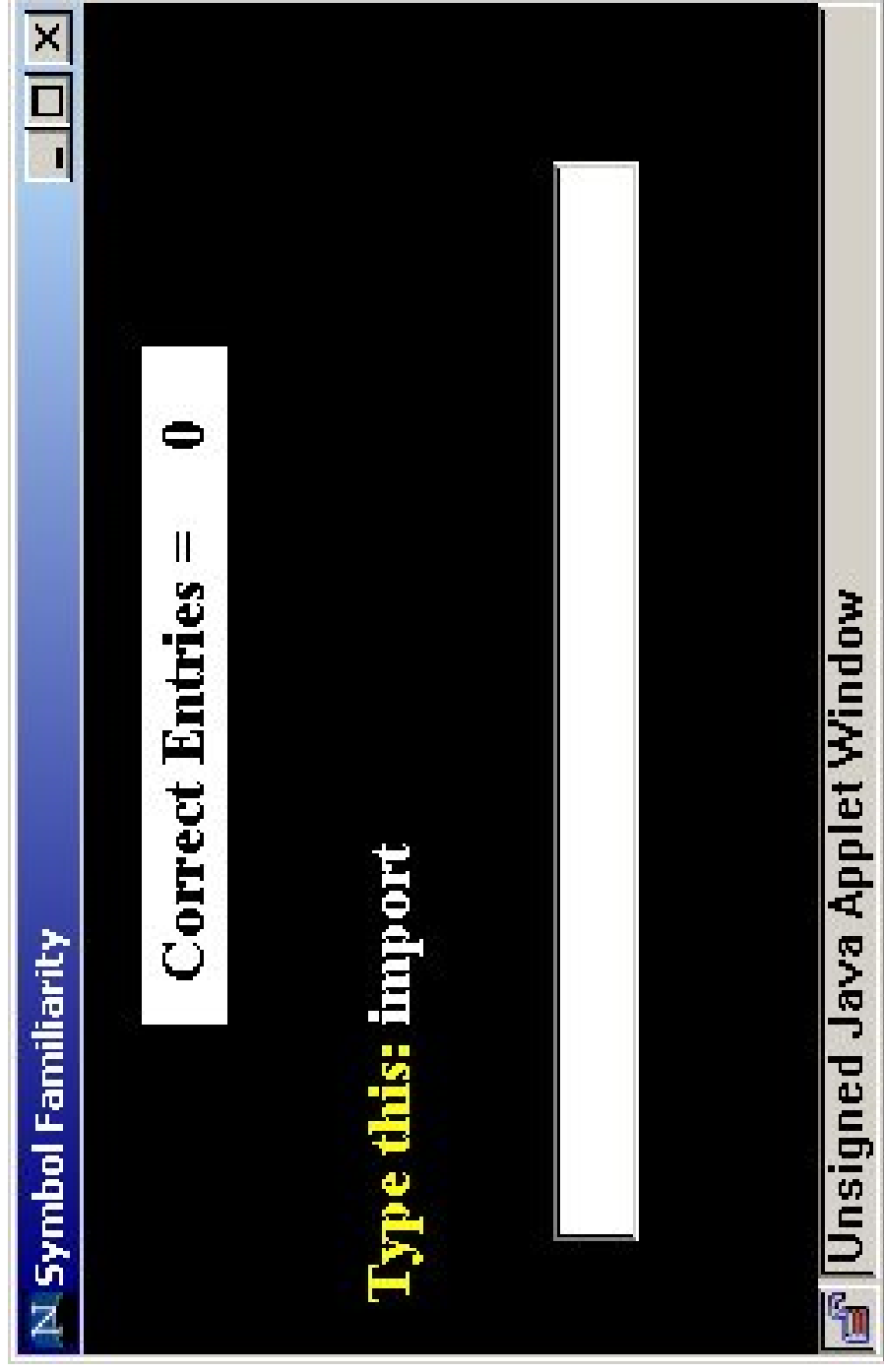
HTML Overview

The screenshot shows a window titled "Unsigned Java Applet Window". The window contains two main areas: a text box on the left and a code editor on the right. The text box contains two paragraphs explaining HTML tags and Java applets. The code editor contains the following HTML code:

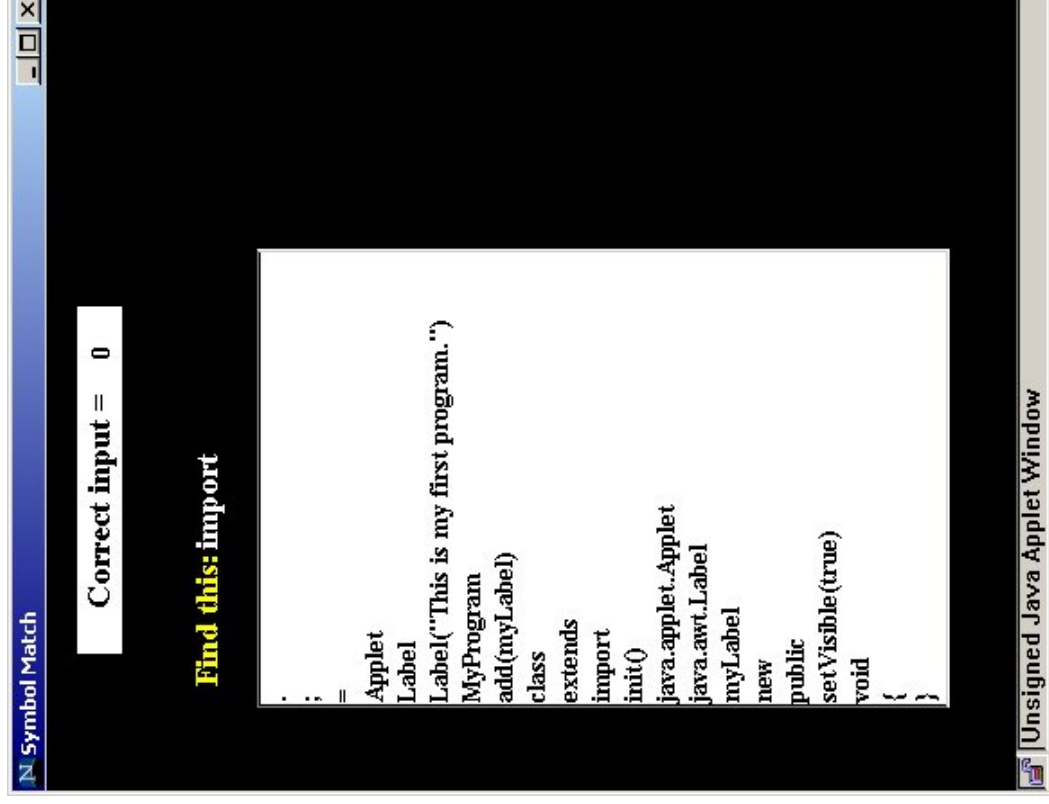
```
<HTML>
<TITLE>MyProgram.HTML</TITLE>
<BODY BGCOLOR=black>
<CENTER>
<APPLET CODE = "MyProgram.class" HEIGHT = 300 WIDTH = 300>
</APPLET>
</CENTER>
</BODY>
</HTML>
```

Below the code editor is a "Proceed..." button. The status bar at the bottom of the window reads "Unsigned Java Applet Window".

Item Familiarity



Item Identification



Item Learning

Learn Java Items - Netscape
File Edit View Go Communicator Help

Loading and initializing may take several seconds to complete. Please wait for the program to start, and do not use the Back or Stop button on the browser.

exposure to new information. You will be more knowledgeable and skilled after even a single repetition of the tutor. And so you are.

As a reminder, here again is the program that you will learn to write and to understand:

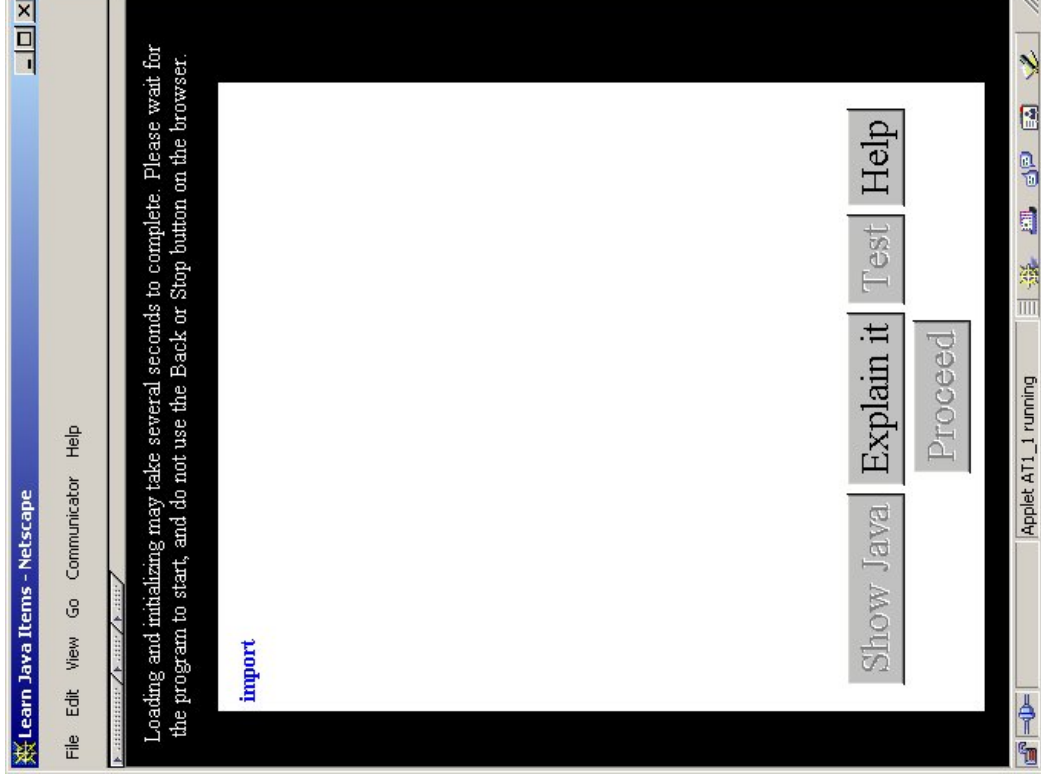
```
import java.applet.Applet;  
import java.awt.Label;  
public class MyProgram extends Applet {  
    Label myLabel;  
    public void init() {  
        myLabel = new Label("This is my first program.");  
        add(myLabel);  
        myLabel.setVisible(true);  
    }  
}
```

You do not need instructions to use this tutor because the events are determined by the enabled buttons and by the accuracy of your typed input and other selections that you will make. You will know when to try to type a Java

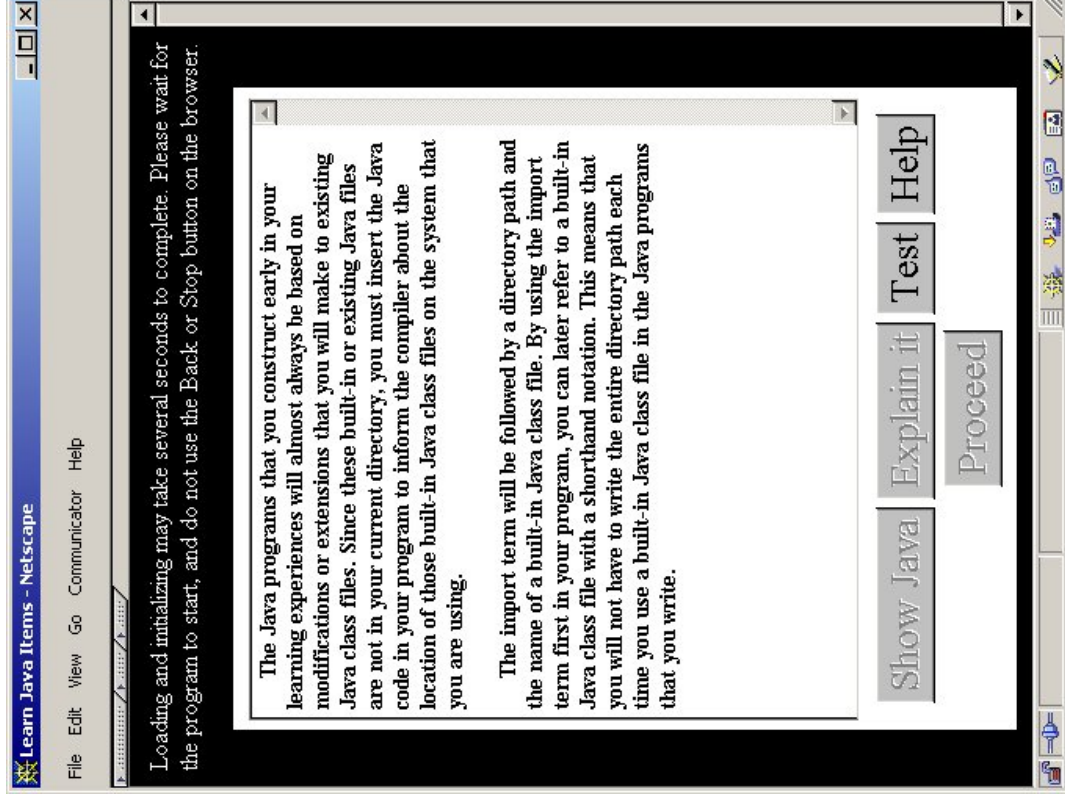
Show Java Explain it Test Help

Proceed

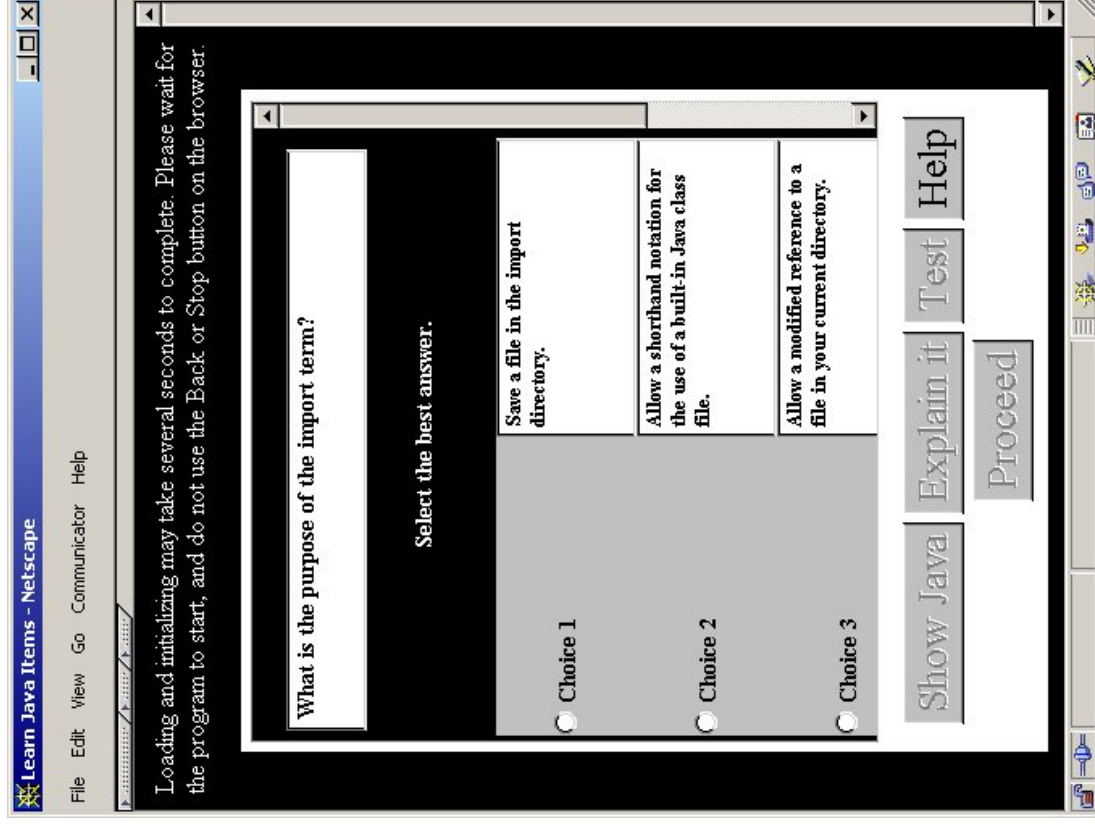
Observe the Item in Context



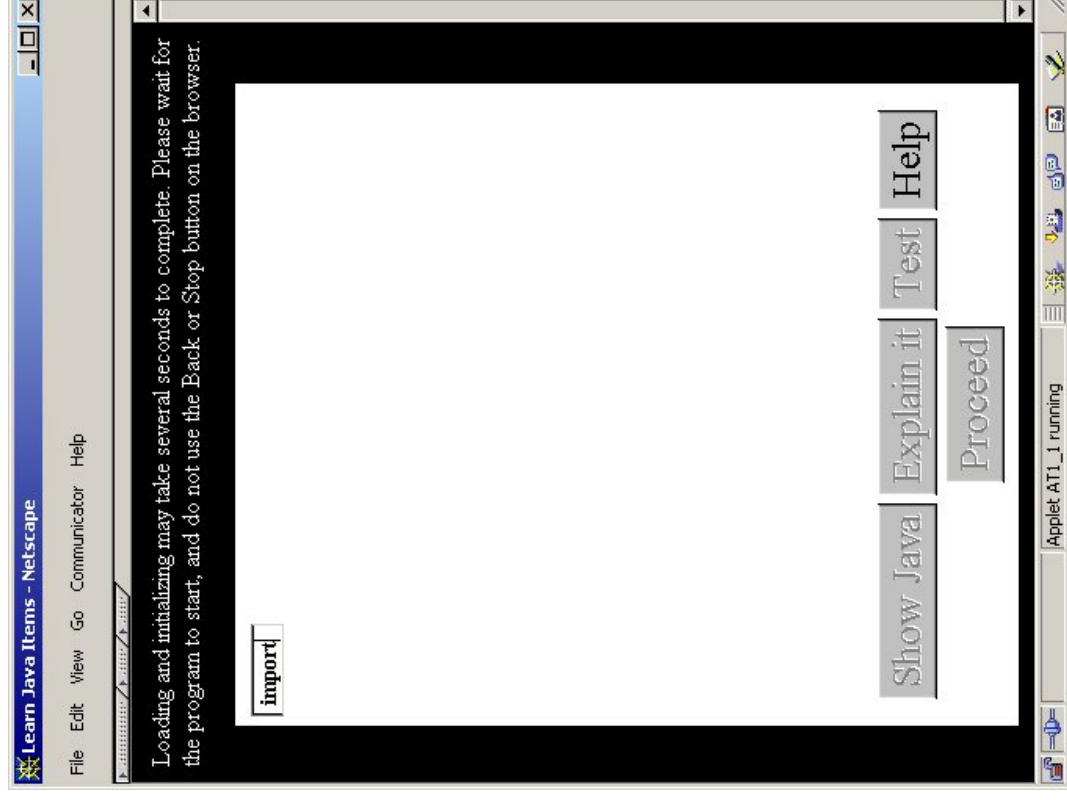
Read the Description (General Rules)



Multiple-Choice Test



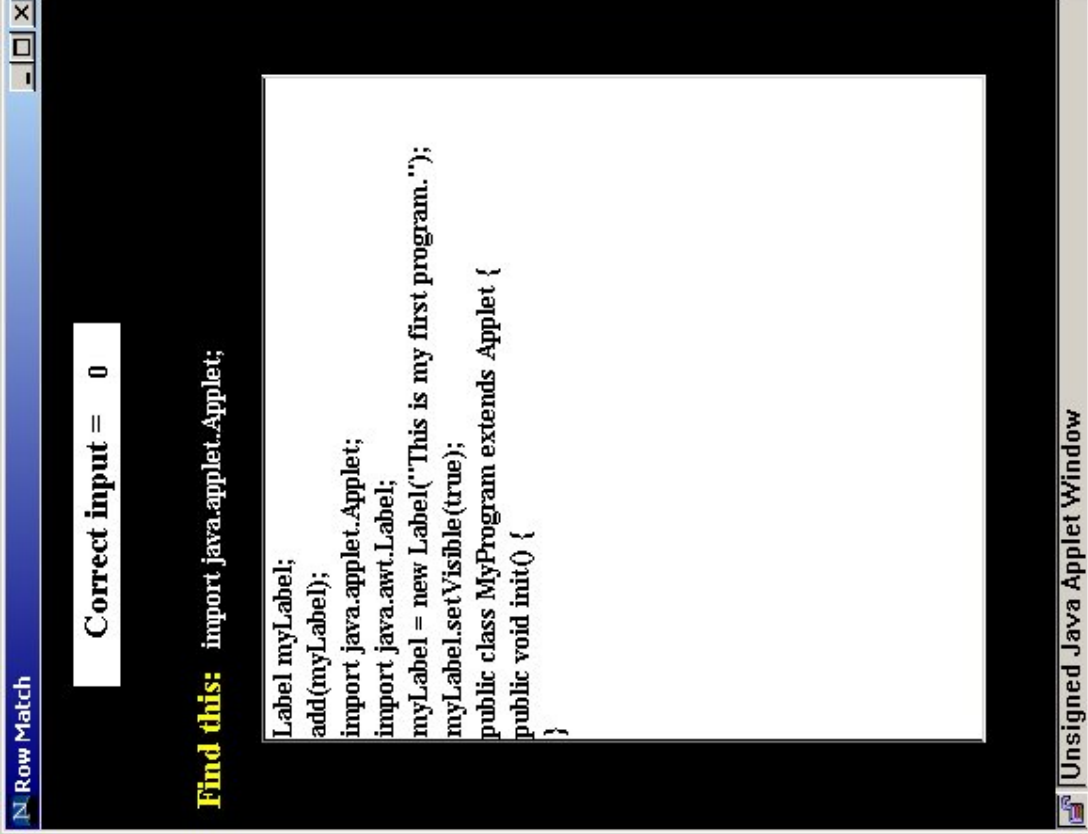
Input the Item from Recall



Row Familiarity



Row Identification



The screenshot shows a Java Applet window titled "Row Match". At the top, there is a text input field containing the text "Correct input = 0". Below the input field, there is a code editor displaying the following Java code:

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

Find this: `import java.applet.Applet;`

Row Interface

Java Tutoring System: Pass 1 of 3

Row 1	
Row 2	
Row 3	
Row 4	
Row 5	
Row 6	
Row 7	
Row 8	
Row 9	
Row 10	

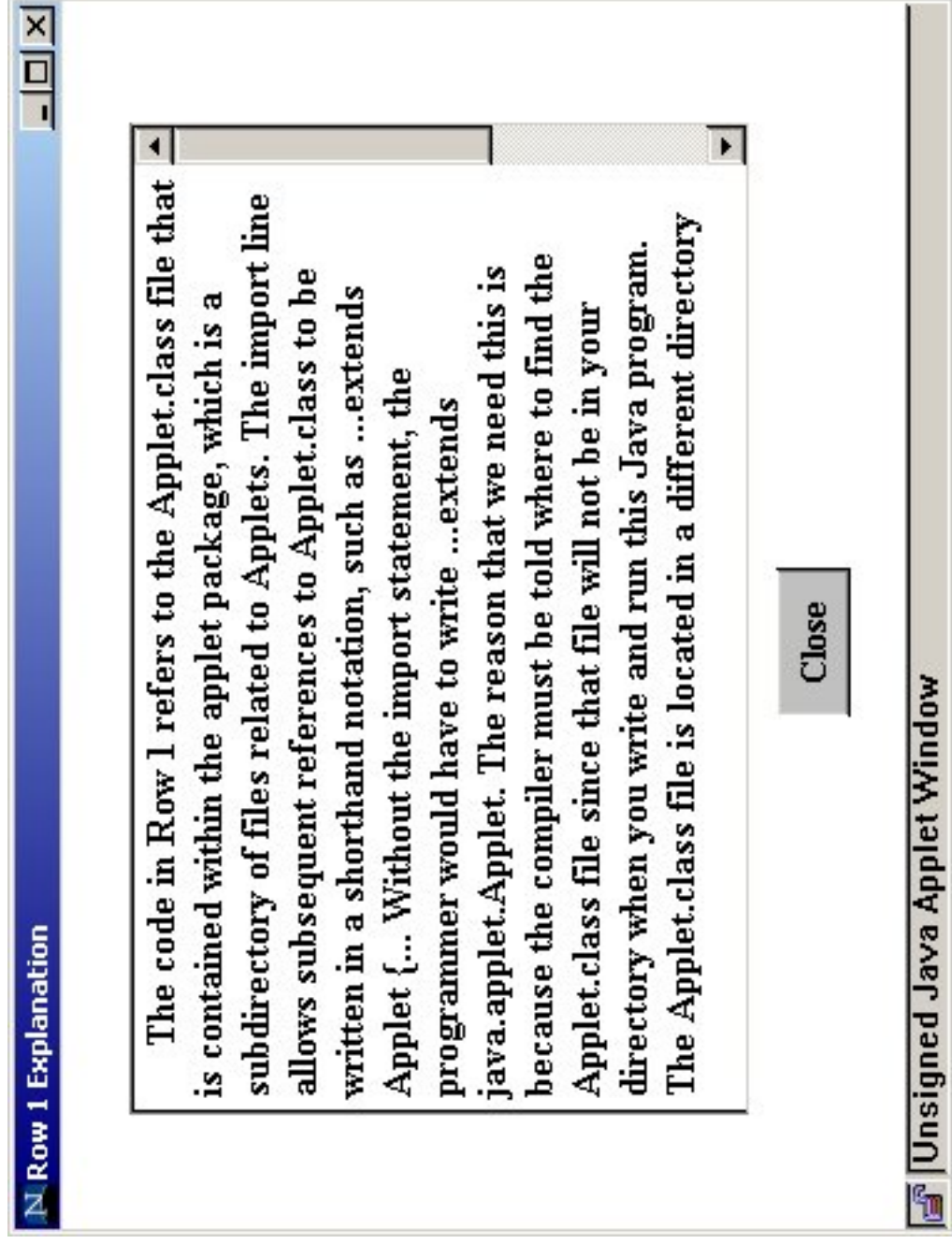
Explain the code?

Unsigned Java Applet Window

Row Interfaces

- **Pass 1**
 - Similar to Item interface
 - Observe the code
 - Read the meaning of a row
 - Take a multiple-choice test after correct input
- **Pass 2**
 - Observe the code
 - Repeat input until correct
- **Pass 3**
 - Whenever observe the code on a row, clear all rows, and start over

Explanation of a Row



Observe the Code

The screenshot shows a window titled "Java Tutoring System: Pass 1 of 3". The main area is a code editor with 10 rows. Row 1 contains the code `import java.applet.applet;`. The other rows are empty. A red button labeled "Show the code?" is located at the bottom right of the editor area. The status bar at the bottom of the window reads "Unsigned Java Applet Window".

Row 1	<code>import java.applet.applet;</code>
Row 2	
Row 3	
Row 4	
Row 5	
Row 6	
Row 7	
Row 8	
Row 9	
Row 10	

Take Test After Correct Input

The screenshot shows a window titled "Java Tutoring System: Pass 1 of 3". The main area contains a code editor with 10 rows. The first three rows contain the following code:

```
Row 1 import java.applet.Applet;  
Row 2 import java.awt.Label;  
Row 3 public class MyProgram extends Applet {  
Row 4  
Row 5  
Row 6  
Row 7  
Row 8  
Row 9  
Row 10
```

A yellow "Take Test" button is located in the top right corner of the code editor area. At the bottom of the window, there is a status bar that reads "Unsigned Java Applet Window".

Text Window

The screenshot shows a window titled "Unsigned Java Applet Window". Inside the window, there is a text area containing 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);  
    }  
}
```

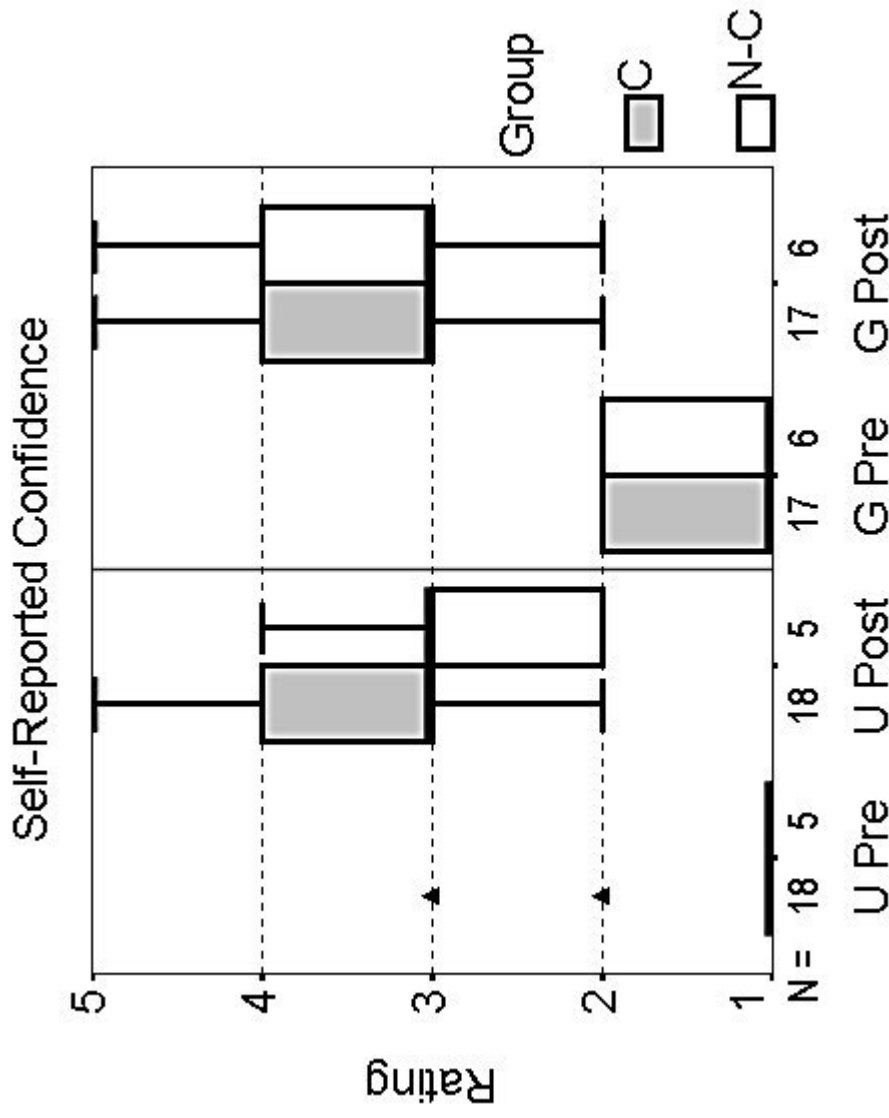
Below the text area, there are two buttons: "Submit" and "Clear".

At the top of the window, there is a blue header bar with the text "Text Editor Emulation".

At the bottom of the window, there is a status bar with the text "Unsigned Java Applet Window".

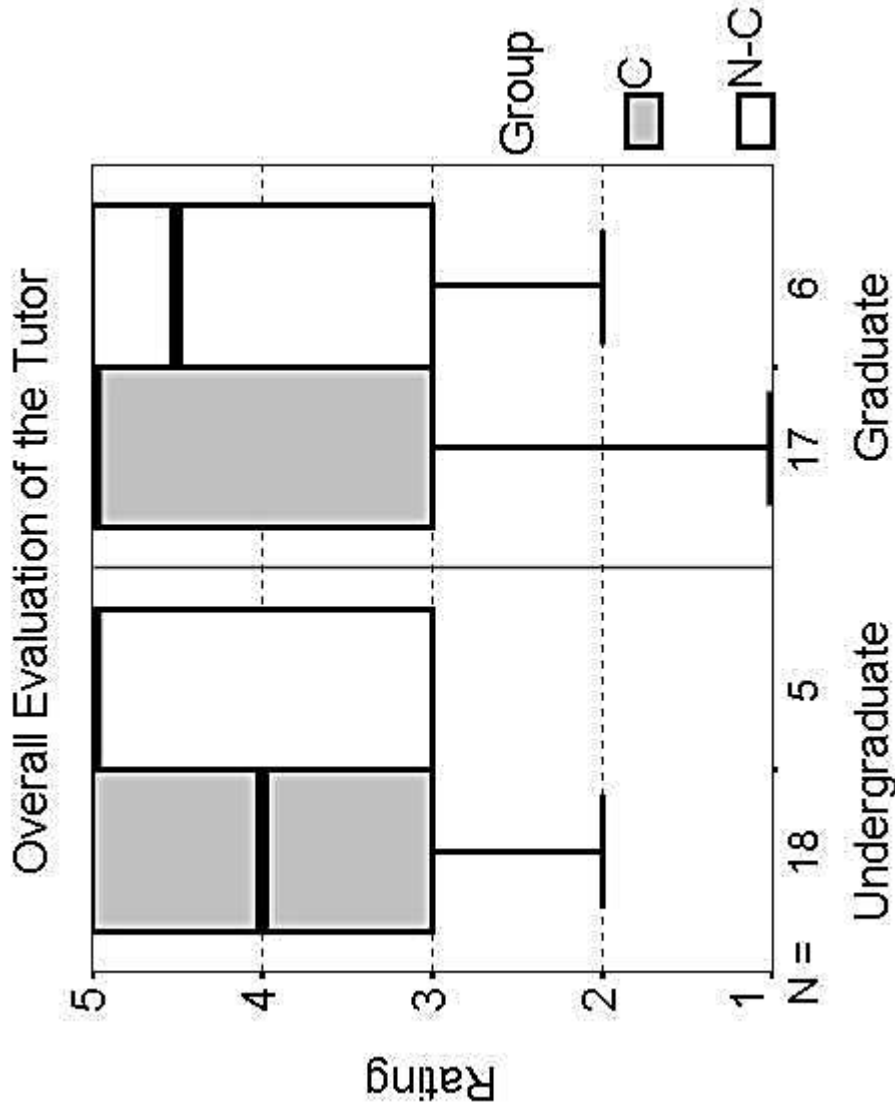
Instructions at the top of the window read: "Type the program in the white space below, and select the Submit button. Do not use the Tab key. Please do not use notes. Try to enter the program from your memory. If you can't remember the program, just select Submit, and you can see the code again."

Confidence in Using the Items of Code



1 = Not at all confident. I do not know how to use the symbol.
 5 = Totally confident. I know how to use the symbol.

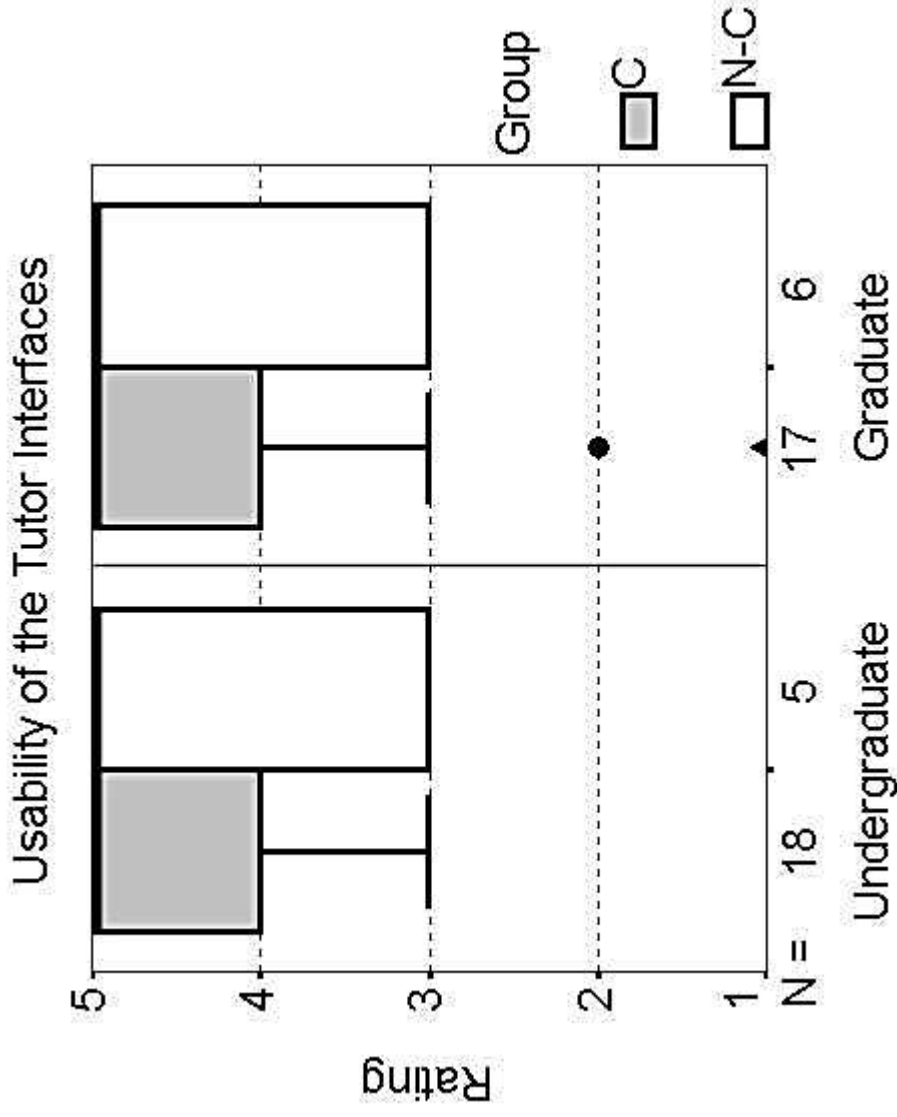
Overall Evaluation of the Tutor



1 = Totally negative. I did not like the tutor.

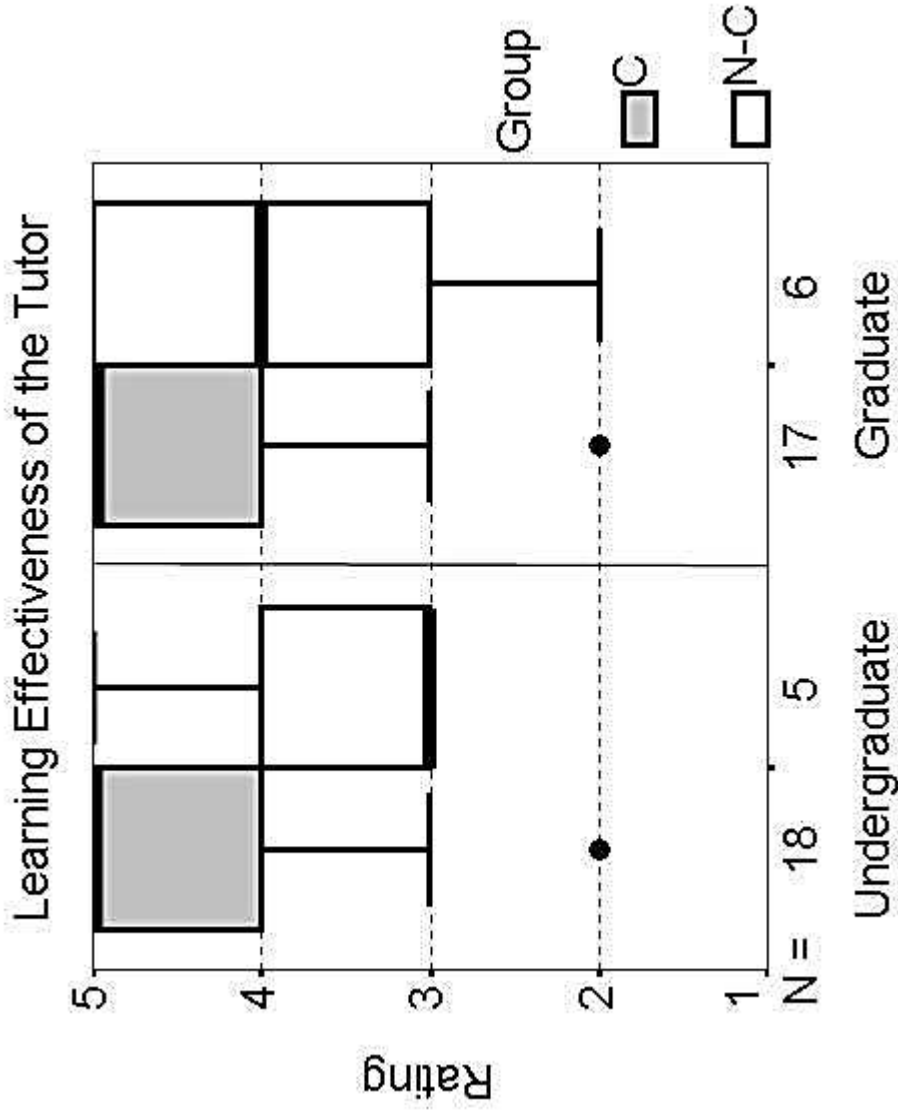
5 = Totally positive. I liked the tutor.

Usability of the Tutor Interfaces



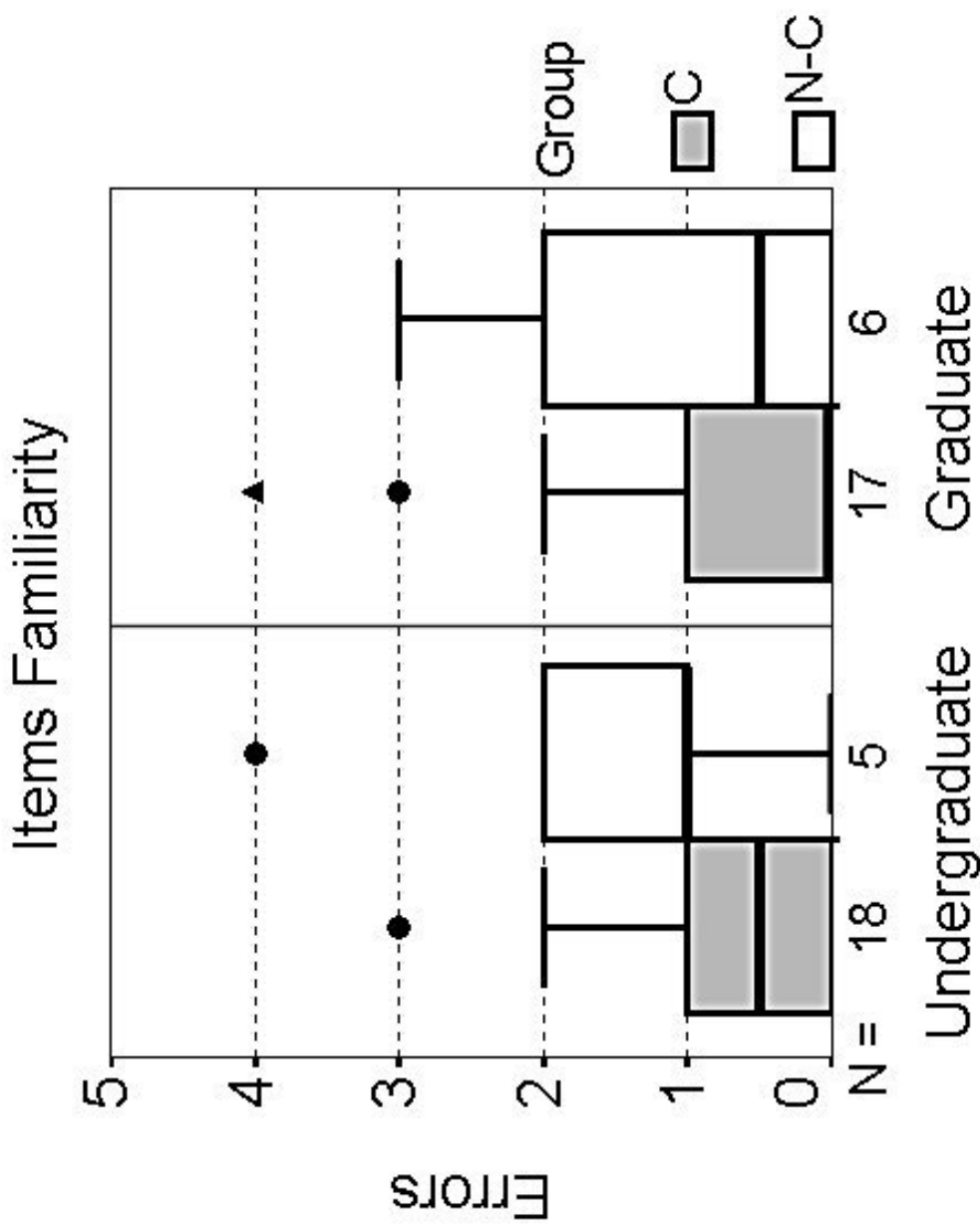
1 = Totally negative. The tutor was difficult to use.
5 = Totally positive. The tutor was easy to use.

Tutor Effectiveness to Learn Java

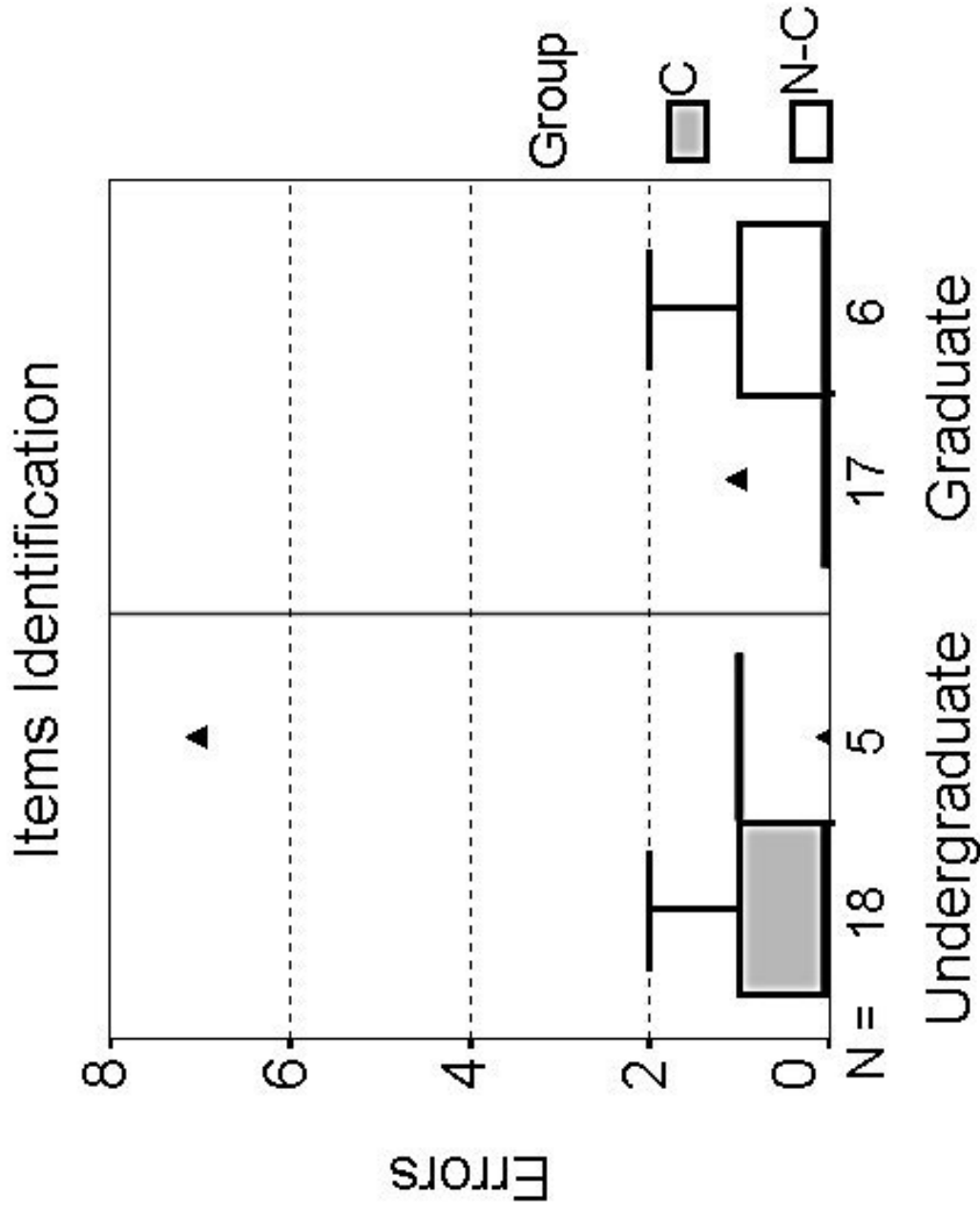


1 = Totally negative. The tutor did not help me to learn Java.
5 = Totally positive. The tutor did help me to learn Java.

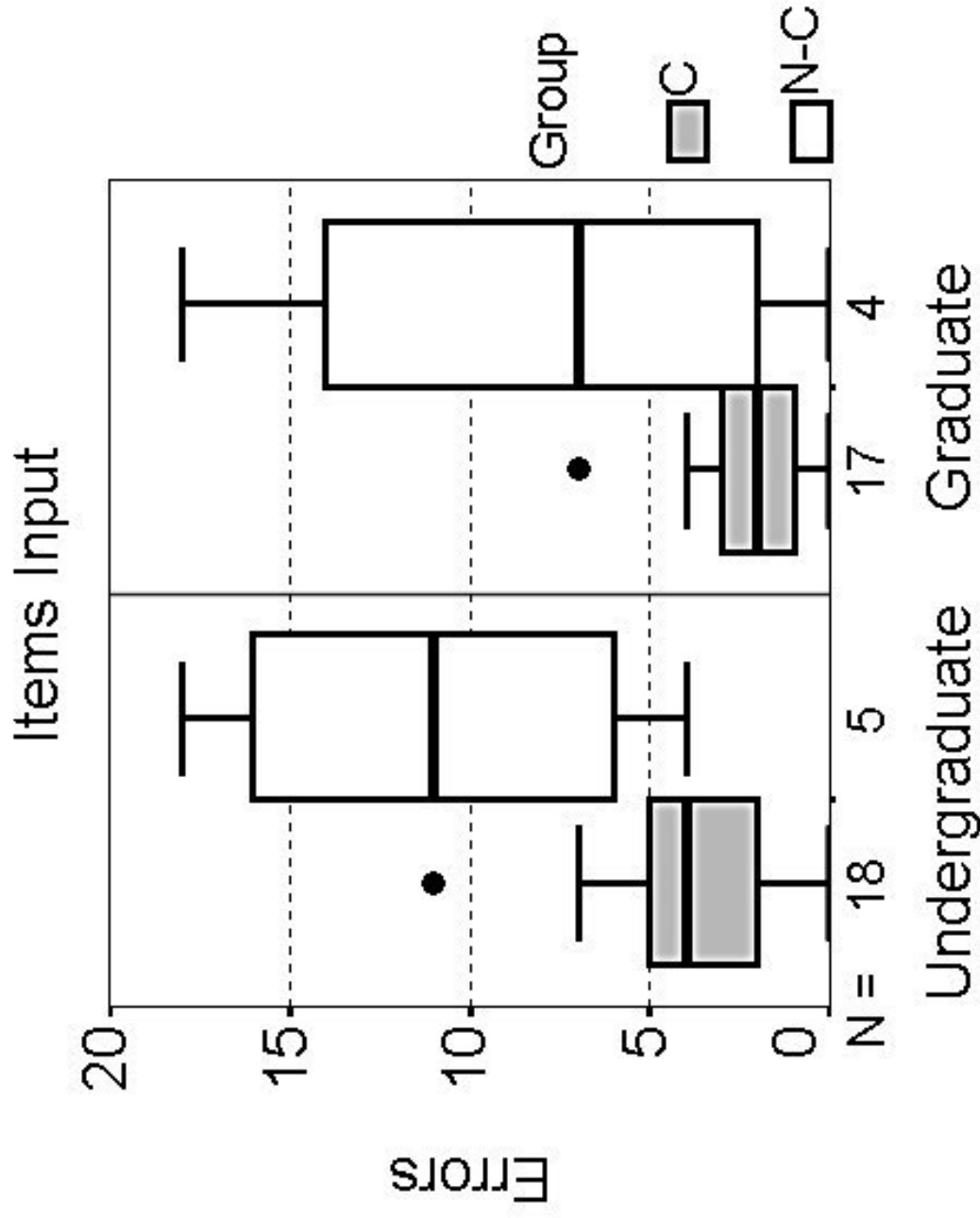
Items Familiarity Performance



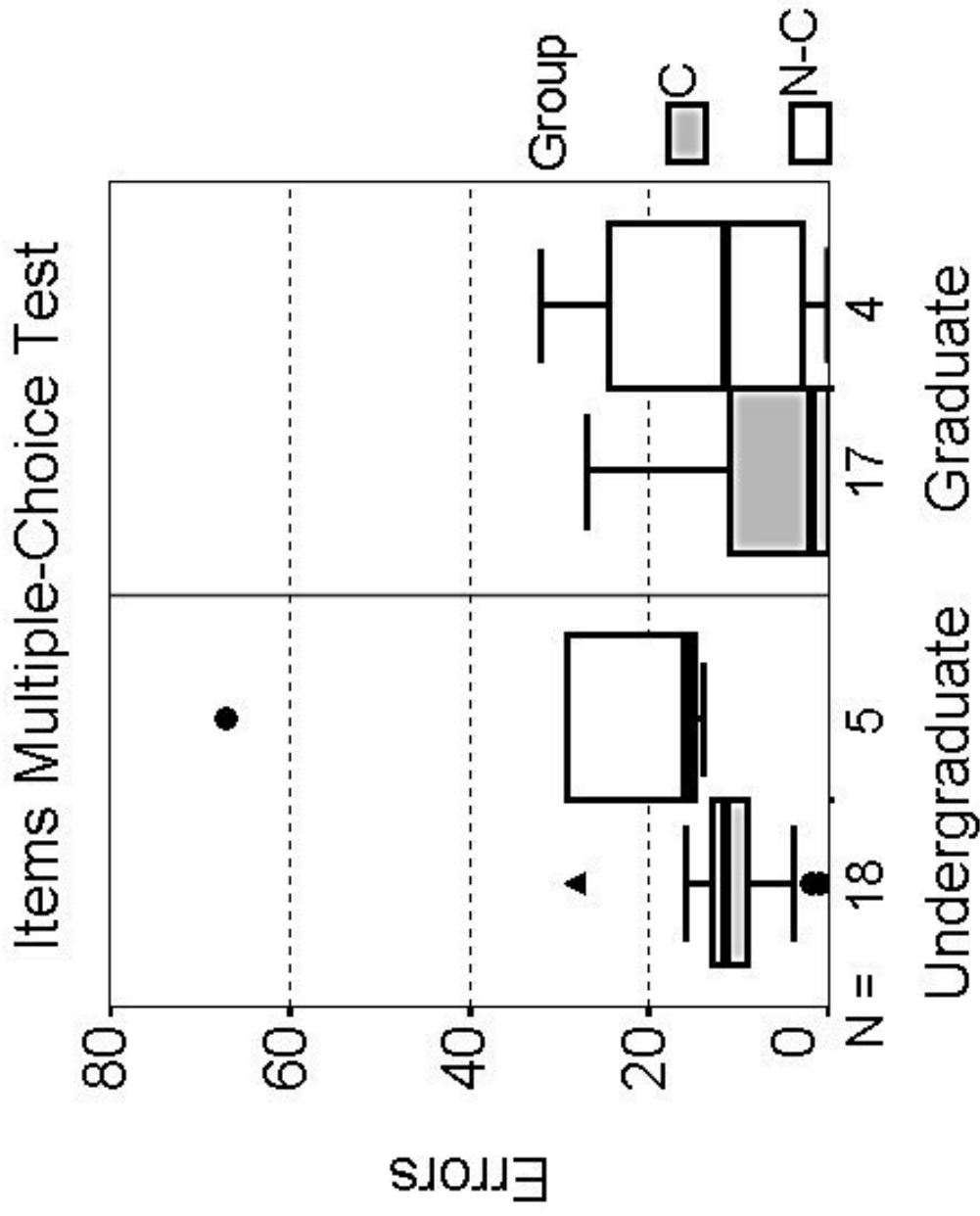
Items Identification Performance



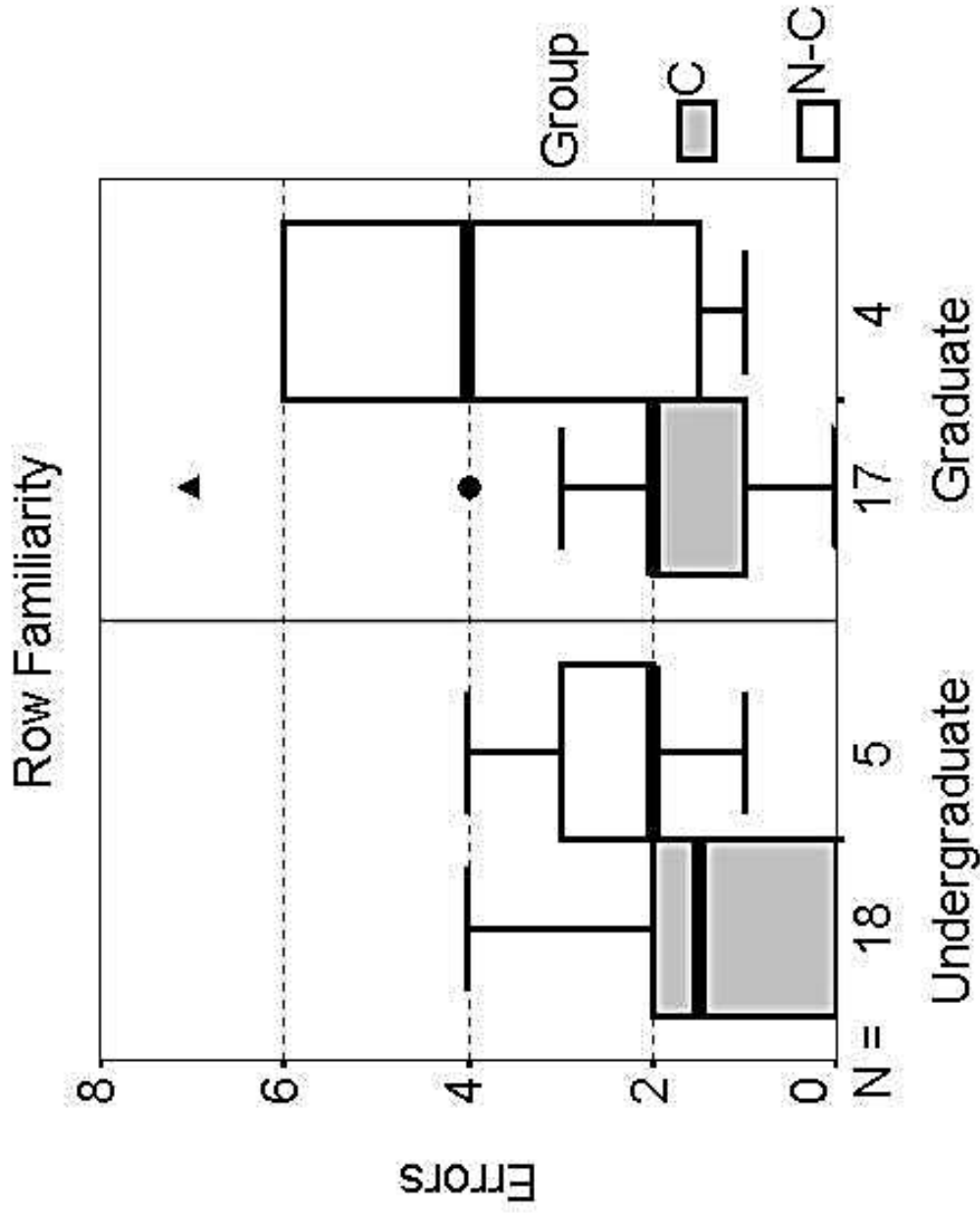
Items Input Performance (Errors)



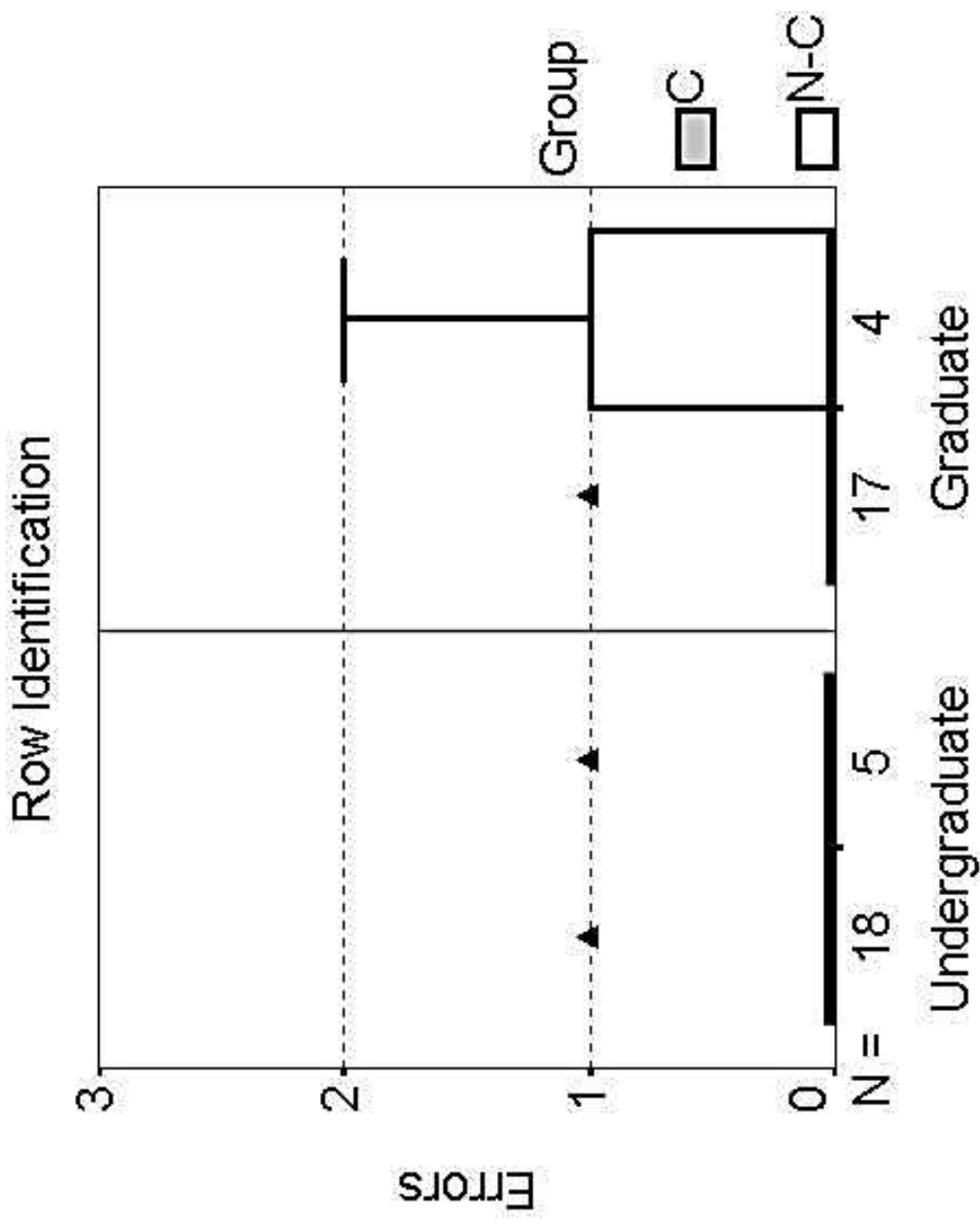
Items Test Performance (Errors)



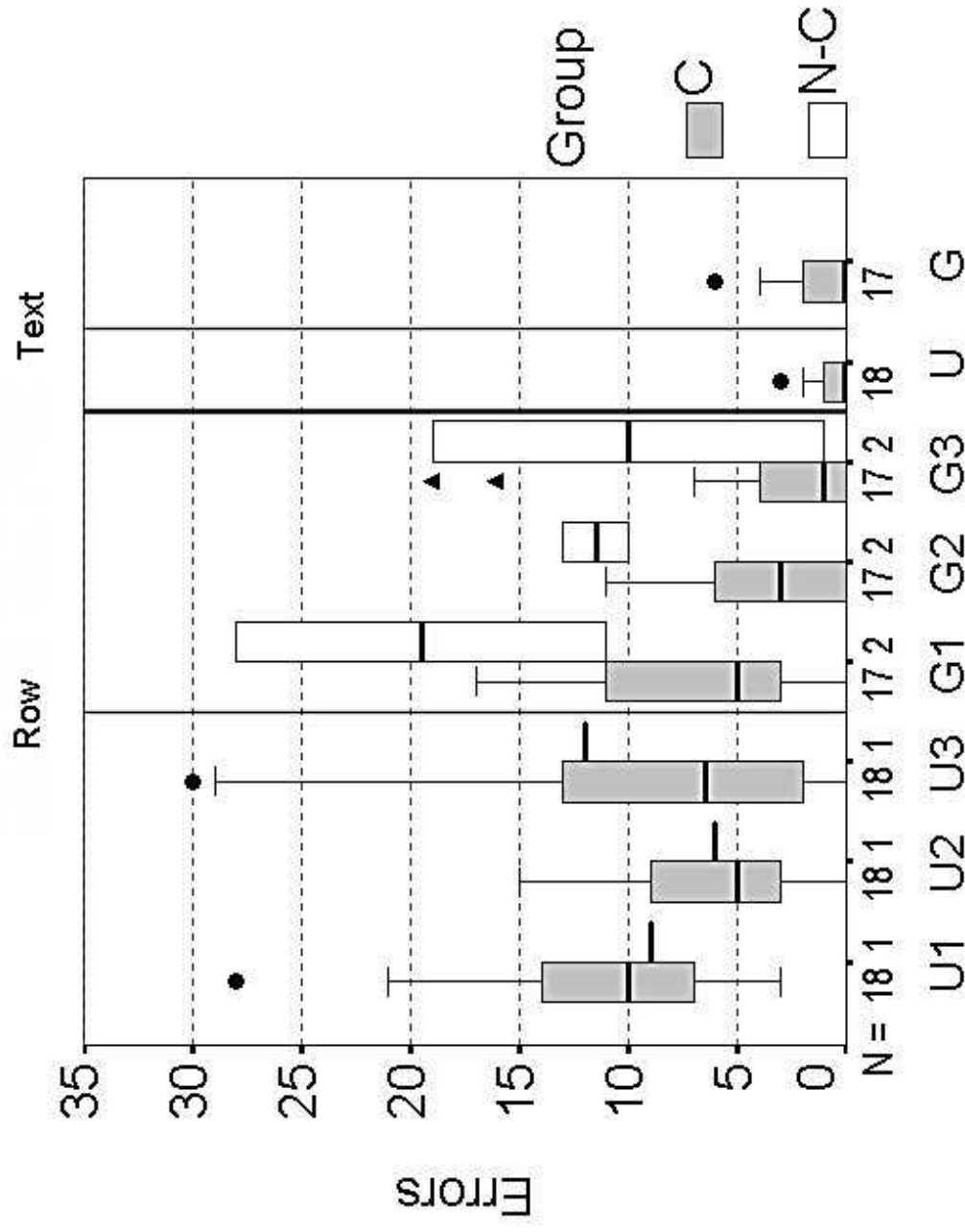
Row Familiarity



Row Identification

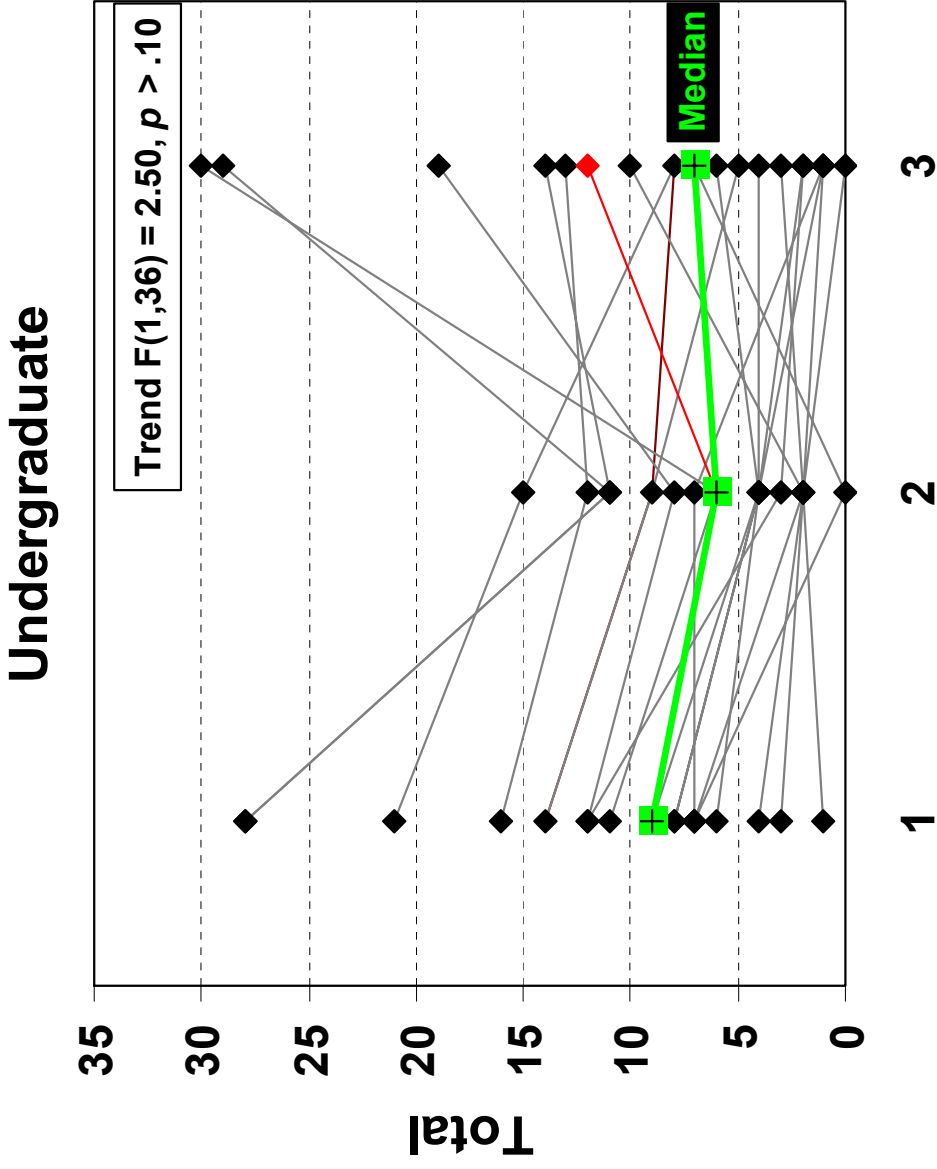


Row and Text Input Performance (Errors)



Row Input Errors: Undergraduate

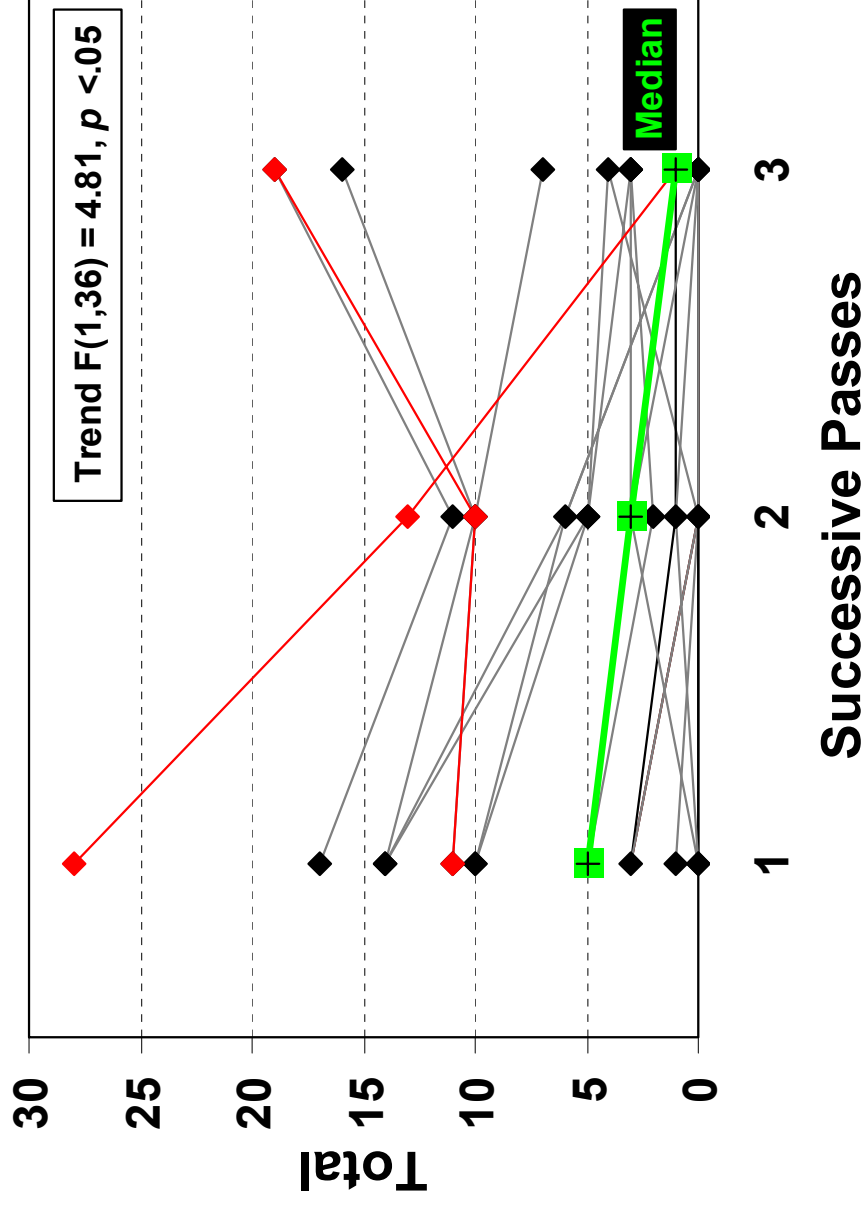
Row Interface Input Errors ($n = 19$)



Successive Passes

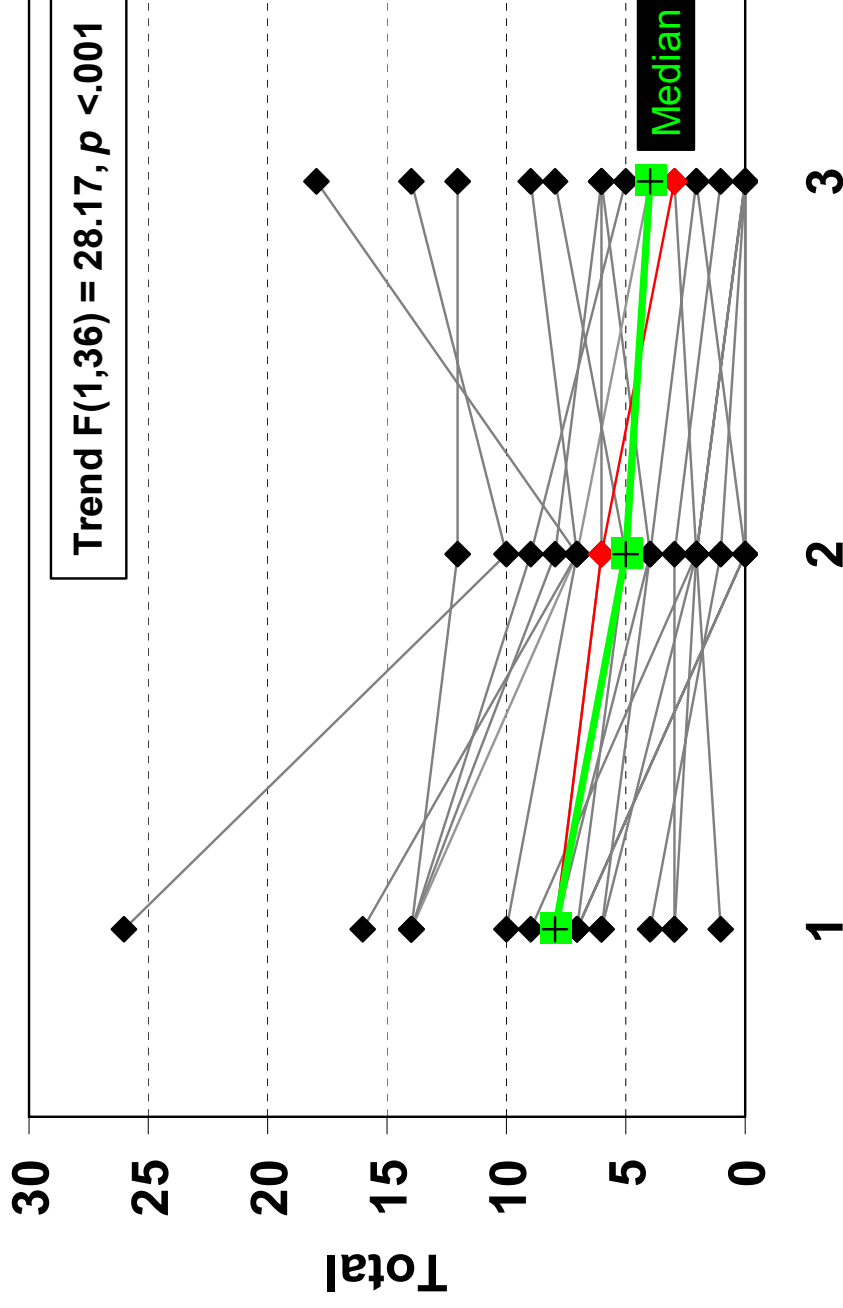
Row Input Errors: Graduate

Row Interface Input Errors ($n = 19$) Graduate



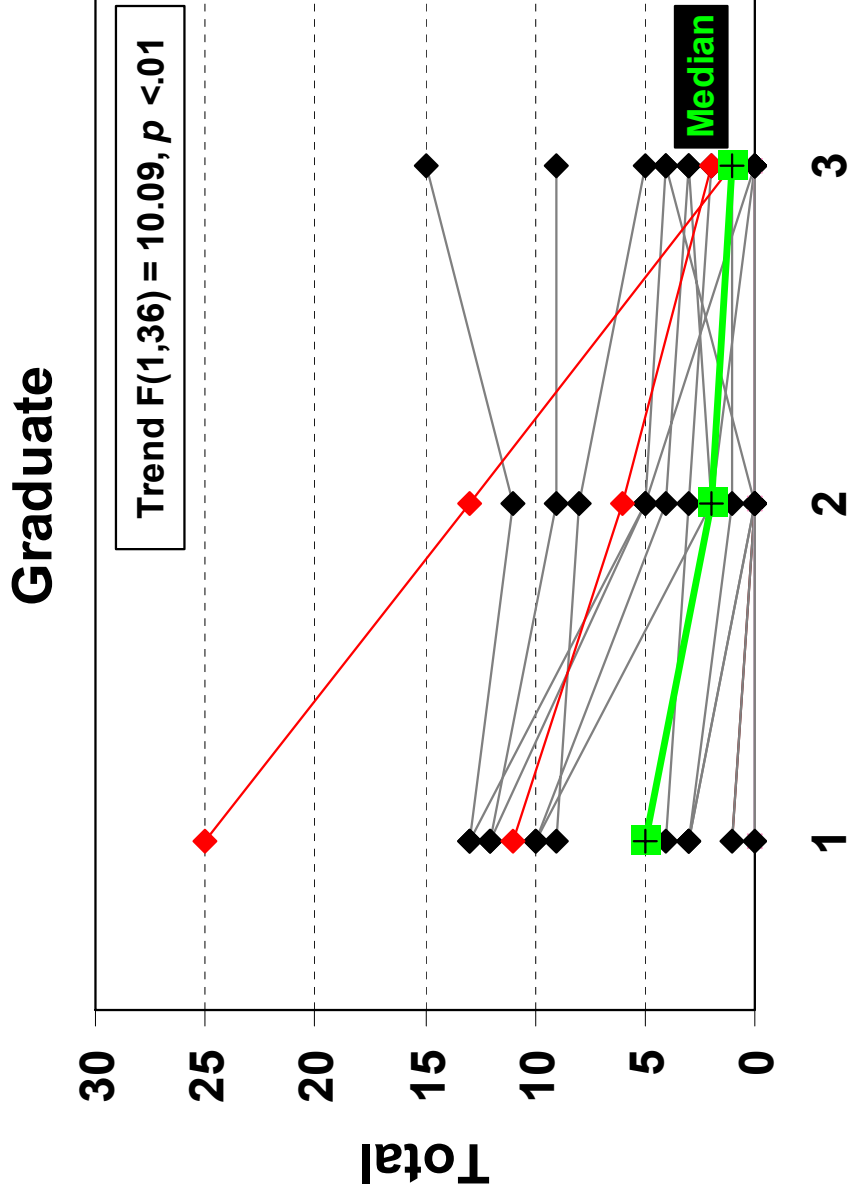
Show Code on a Row: Undergraduate

Row Interface Show Code ($n = 19$)
Undergraduate

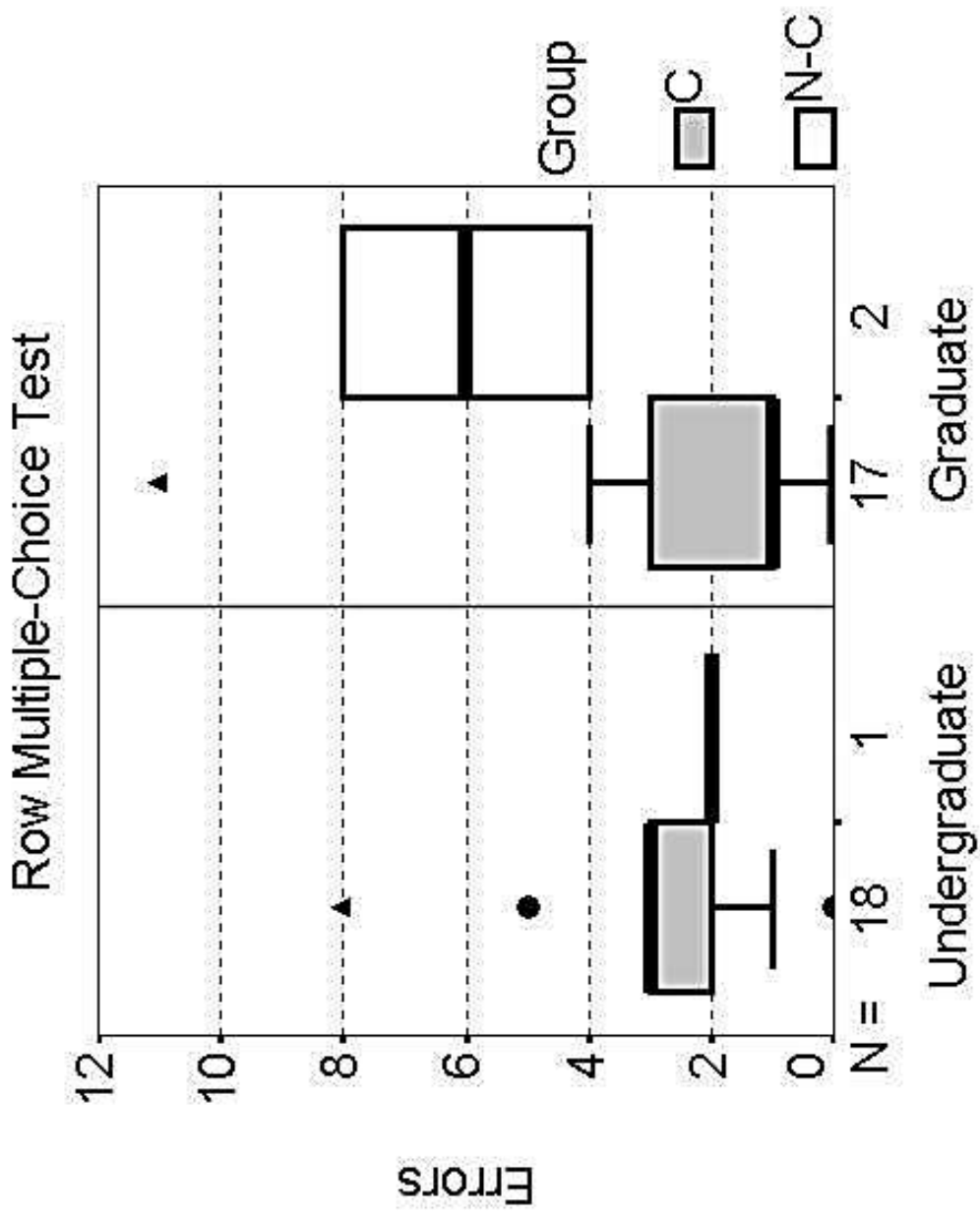


Show Code on a Row: Graduate

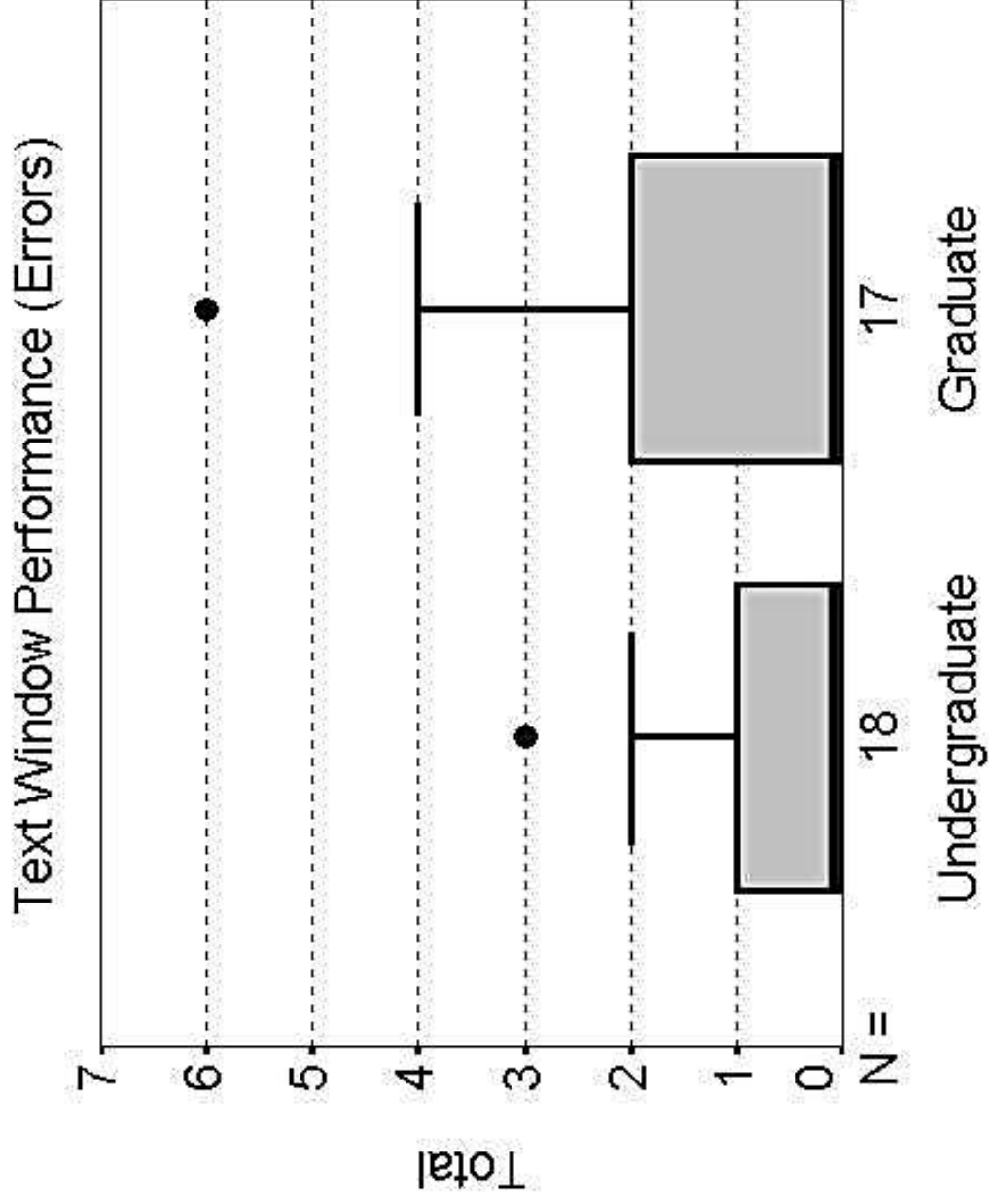
Row Interface Show Code ($n = 19$)



Row Multiple-Choice Test



Text Window Performance



Next class period

- Modified Personalized System of Instruction
 - Lecture
 - Collaboration
- Run the Applet

Conclusions

1. **Structured rehearsal** is effective.
2. **Repetition** is an undervalued factor in learning and retention.
3. The tutor generated opportunities for **overlearning**.
4. Providing a **successful learning experience** early prepares and motivates the student to handle advanced programming techniques taught in conventional ways.

Tutor URL

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