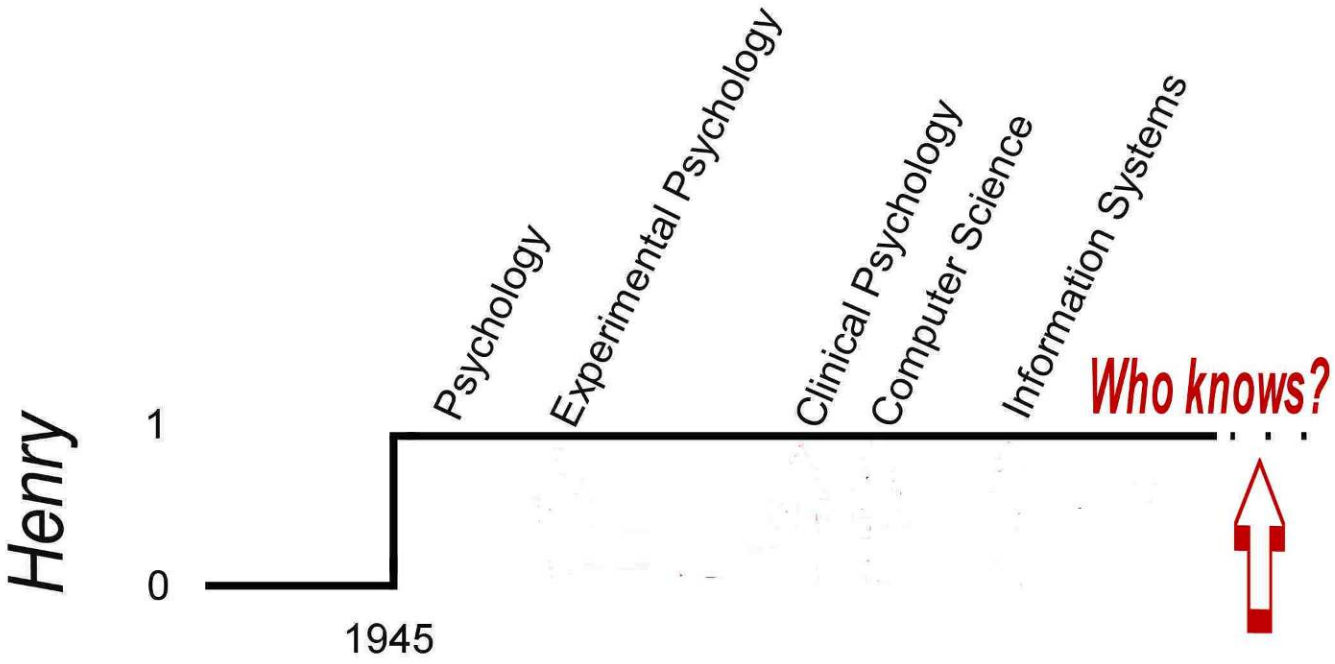
Programmed Instruction Applications to Technology Education

Miji Mathews, Jingli Wang,
Amy Hu, Valeri Scott,
John Goodall, Xin Li, & Diana Wang
UMBC
&
Ashley G. Durham
Centers for Medicare and Medicaid Services
&
Henry H. Emurian
Information Systems Department
UMBC

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Cumulative record of Henry



UMBC
AN HONORS UNIVERSITY IN MARYLAND

I am right.

The organism is always right…

How it all started (circa 1982)

```
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
```

UMBC
AN HONORS UNIVERSITY IN MARYLAND

20 Years Later (circa 2002)

```
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. }
```

UMBC
AN HONORS UNIVERSITY IN MARYLAND

What I used to do at UMBC

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

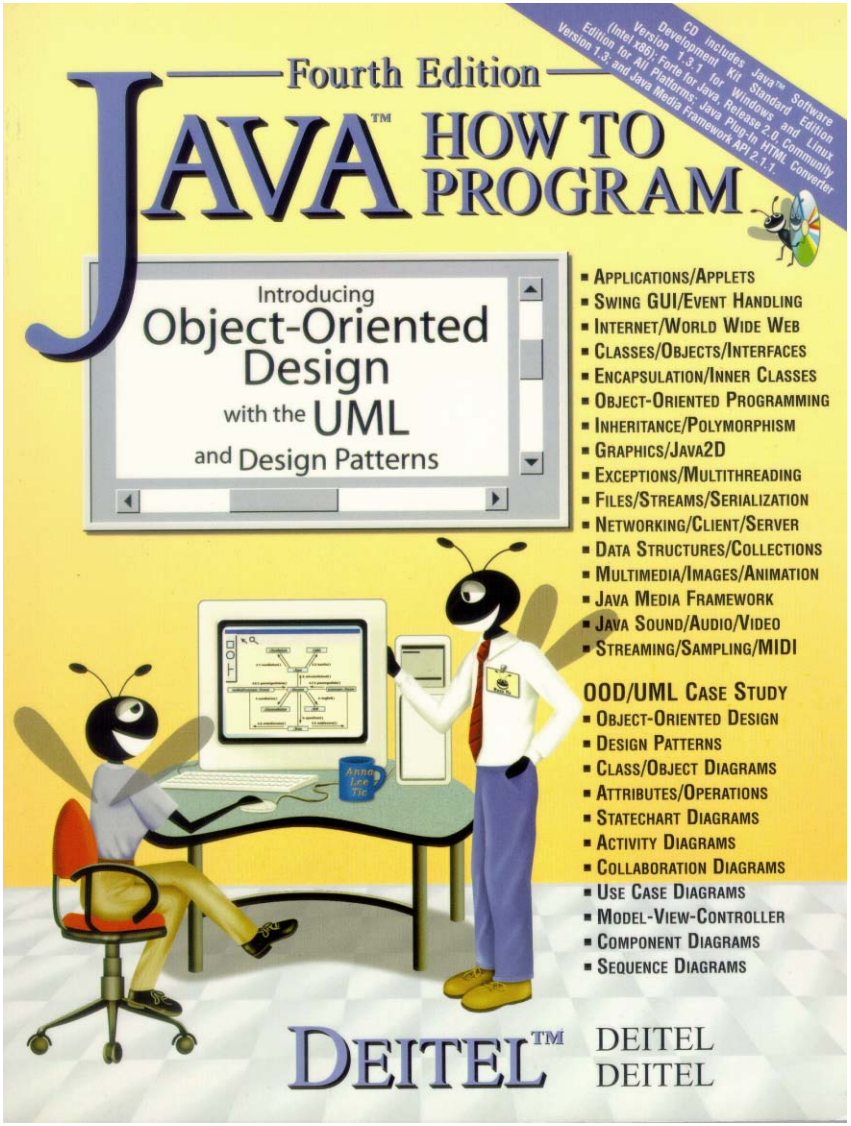
UMBC
AN HONORS UNIVERSITY IN MARYLAND

What I noticed (The end of denial...)

- ***Student "Motivation"***
  - **Information Systems students are less keen to study computer programming than are computer science students.**
  - **Students are *insensitive to reinforcers* that are symbols.**
  - **Students lack a history that produces *generalized conditioned reinforcers* that can sustain learning.**

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Prentice Hall, 2002



TERMINOLOGY

- applet
- applet container
- `<applet>` tag
- **Applet** menu
- **appletviewer**
- **boolean** primitive type
- browser
- built-in data type
- **byte** primitive type
- **char** primitive type
- command-line argument
- coordinate
- create an object
- derived class
- **double** primitive data type
- **Double.parseDouble** method
- double-precision floating-point number
- **drawLine** method of class **Graphics**
- **drawRect** method of class **Graphics**
- **drawString** method of class **Graphics**
- **extends** keyword
- **float** primitive type
- floating-point number
- **Graphics** class
- **height** of an applet
- HTML Converter
- HTML tag
- Hypertext Markup Language (HTML)
- import statement
- information hiding
- **init** method of class **JApplet**
- instance variable
- instantiate an object
- **int** primitive type
- interface
- invoke a method
- **JApplet** class
- **java.awt** package
- Java Plug-in
- Java 2 Runtime Environment (J2RE)
- **javax.swing** package
- local variable
- logic error
- **long** primitive type
- message
- method call
- Microsoft Internet Explorer
- Netscape Communicator
- object
- **paint** method of class **JApplet**
- parameter list
- pixel (picture element)
- primitive data type
- **Quit** menu item
- references
- **Reload** menu item
- **short** primitive type
- single-precision floating-point number
- source code
- **start** method of class **JApplet**
- subclass
- superclass
- text file
- **width** of an applet
- World Wide Web

UMBC
AN HONORS UNIVERSITY IN MARYLAND

SELF-REVIEW EXERCISES

3.1 Fill in the blanks in each of the following.

a) Class _____ provides methods for drawing.

b) Java applets begin execution with a series of three method calls: _____, _____ and _____.

c) Methods _____ and _____ display lines and rectangles.

d) Keyword _____ indicates that a new class is a subclass of an existing class.

e) Every Java applet should extend either class _____ or class _____.

f) Java's eight primitive data types are _____, _____, _____, _____, _____, _____, _____ and _____.

3.2 State whether each of the following is *true* or *false*. If *false*, explain why.

a) To draw a rectangle, method **`drawRect`** requires four arguments that specify two points on the applet.

b) Method **`drawLine`** requires four arguments that specify two points on the applet to draw a line.

c) Type **`Double`** is a primitive data type.

d) Data type **`int`** is used to declare a floating-point number.

e) Method **`Double.parseDouble`** converts a **`String`** to a primitive **`double`** value.

3.3 Write Java statements to accomplish each of the following:

a) Display a dialog asking the user to enter a floating-point number.

b) Convert a **`String`** to a floating-point number and store the converted value in **`double`** variable **`age`**. Assume that the **`String`** is stored in **`stringValue`**.

c) Draw the message **`"This is a Java program"`** on one line on an applet (assume you are defining this statement in the applet's **`paint`** method) at position *(10, 10)*.

d) Draw the message **`"This is a Java program"`** on two lines on an applet (assume these statements are defined in applet method **`paint`**) starting at position *(10, 10)* and where the first line ends with **`Java`**. Make the two lines start at the same *x* coordinate.

What I came to re-value…

- ***Rote memorization*** **is fundamental to the acquisition of skills that set the occasion for "understanding."**
  - **"Constructivism" comes later (much, much later…)**
- ***Disciplined study behavior*** **is essential to acquire skill, and many, if not most students, do not know how to study. They don't know what state is "steady."**

UMBC
AN HONORS UNIVERSITY IN MARYLAND

What I came to re-value, too

- ***Repetition*** **and overlearning are essential to the learning process.**
- ***Feeling good about yourself*** **after hard work sustains enthusiasm for more learning.**

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Regurgitation and beyond the mind

- ***Responding is Good***
  - **Mindful, intentional, informed, purposeful, and accurate responding.**
- ***Rule Governed Behavior***
  - **Within the context of the interactive tutor, frames of information about general syntax and semantics of Java and object-oriented programming are presented.**
  - **Multiple-choice tests based on these frames are embedded throughout the tutor.**

UMBC
AN HONORS UNIVERSITY IN MARYLAND

What I used to do (again)

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

UMBC
AN HONORS UNIVERSITY IN MARYLAND

The Enlightenment!

Problem

```
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. }
```

UMBC
AN HONORS UNIVERSITY IN MARYLAND

The Enlightenment!

Problem

```
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

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

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Programmed Instruction

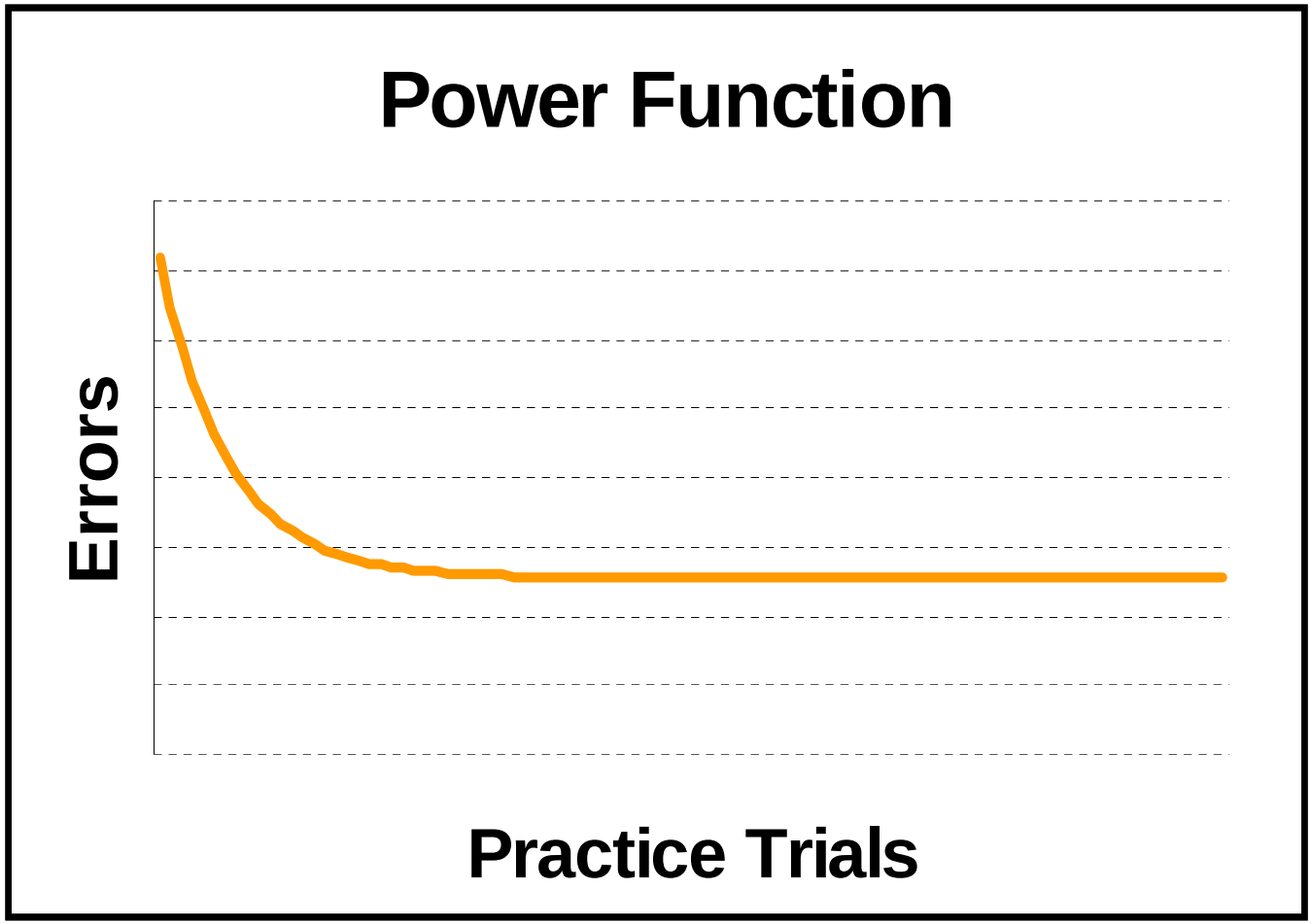
- **A set of *structured interactions* between a learner and the material to be mastered.**
- **Structures study behavior that is focused on the *individual learner*.**
- **Manages the *moment-by-moment interactions* between a learner and a tutor.**
- ***Step-wise progression* from elementary knowledge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

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Acquisition Curve

What state is “steady”?

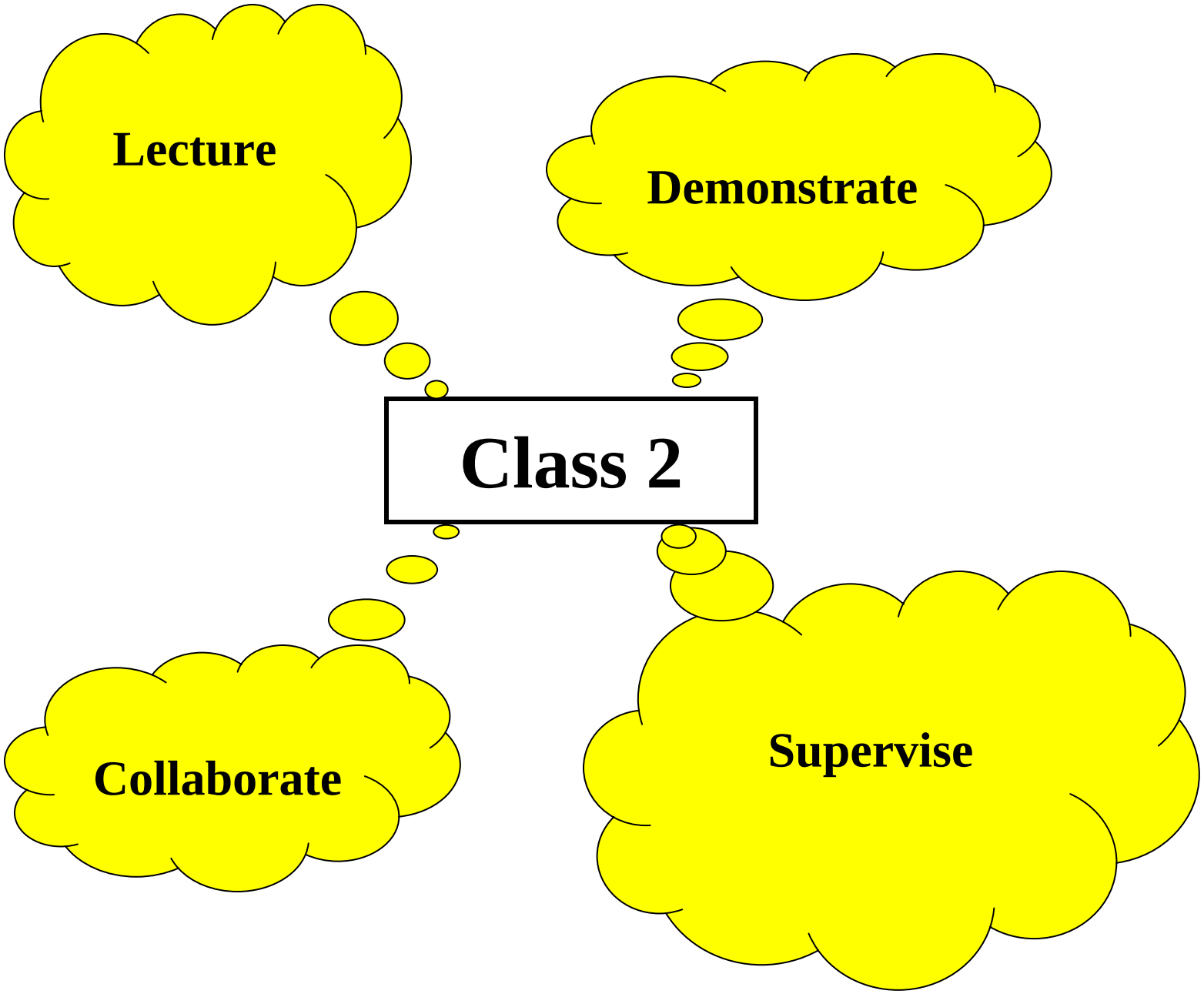
How do you get there and know when you’ve arrived?



UMBC
AN HONORS UNIVERSITY IN MARYLAND

Programmed Instruction

Class 1


Lecture
Demonstrate
Class 2
Collaborate
Supervise

Consultant

Class 14

Self-Regulated Learners
(“Constructivists”)

Tactic: Programmed Instruction

- Behavior Analysis
  - Skinner, B.F. (1954). The Science of Learning and the Art of Teaching, *Harvard Educational Review*, *24*, 86-97.
  - Skinner, B.F. (1958). Teaching machines. *Science*, *128*, 969-977.

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Eleven Features of PI
(Holland, 1960; Scriven, 1969; Skinner, 1958; Vargas & Vargas, 1991)

1. Comprehensibility of each unit or "frame,"
2. Tested effectiveness of a set of frames,
3. Skip-proof frames,
4. Self-correcting tests,
5. Automatic encouragement for learning,
6. Diagnosis of misunderstandings,
7. Adaptations to errors by hints, prompts, and suggestions, cont'd

Features of PI

9. Learner constructed responses based on recall,
10. Immediate feedback, successive approximations to a terminal objective, and
11. Student-paced progress.

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Down Memory Lane...

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Socrates (cf. Keller)

Meno

By Plato

Written 380 BC
Translated by Benjamin Jowett

Persons of the Dialogue
MENO
SOCRATES
A SLAVE OF MENO
ANYTUS

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Volume 1 Number 1 May 1964

Programmed Learning

JOURNAL OF THE ASSOCIATION FOR PROGRAMMED LEARNING

Editor
JOHN ANNETT

7/6 net 21/- ann.

Sweet & Maxwell

Volume 4 Number 2 April 1967

PROGRAMMED LEARNING & EDUCATIONAL TECHNOLOGY

JOURNAL OF THE ASSOCIATION FOR PROGRAMMED LEARNING

Editor
JOHN ANNETT

17/6 net £3 ann.

Sweet & Maxwell

FEBRUARY 1989
VOLUME 26
NUMBER 1

EDUCATIONAL & TRAINING TECHNOLOGY INTERNATIONAL

Formerly Programmed Learning and Educational Technology

The Journal of the Association for Educational and Training Technology

KOGAN PAGE

EBRUARY 1995
VOLUME 32
NUMBER 1

Innovations in Education and Training International

Formerly published as
Educational and Training Technology International

THE JOURNAL OF
the Association for Educational and Training Technology and the Staff and Educational Development Association.

ISSN 1355 8005

KOGAN PAGE

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

Table 1. A set of frames designed to teach a third- or fourth-grade pupil to spell the word *manufacture*.

1. **Manufacture** means to make or build. *Chair factories manufacture chairs.* Copy the word here:

   □□□□□□□□□□□

2. Part of the word is like part of the word **factory.** Both parts come from an old word meaning *make* or *build.*

   m a n u □□□□ **u r e**

3. Part of the word is like part of the word **manual.** Both parts come from an old word for *hand.* Many things used to be made by hand.

   □□□□ **f a c t u r e**

4. The same letter goes in both spaces:

   m □ **n u f** □ **c t u r e**

5. The same letter goes in both spaces:

   m a n □ **f a c t** □ **r e**

6. **Chair factories** □□□□□□□□□□□ **chairs.**

does not lend itself to excerpting. The 10,000 or 15,000 responses made by each

Pressey's Teaching Machines

There is another kind of capital equipment which will encourage the student to take an active role in the instructional process. The possibility was recognized in the 1920's, when Sidney L. Pressey designed several machines for the automatic testing of intelligence and information. A recent model of one of these

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

with a nicety which no teacher could well equal. The present paper describes this drill machine.

How the Apparatus appears to the Learner

Chart I is an outline sketch of the way the apparatus appears to the subject.

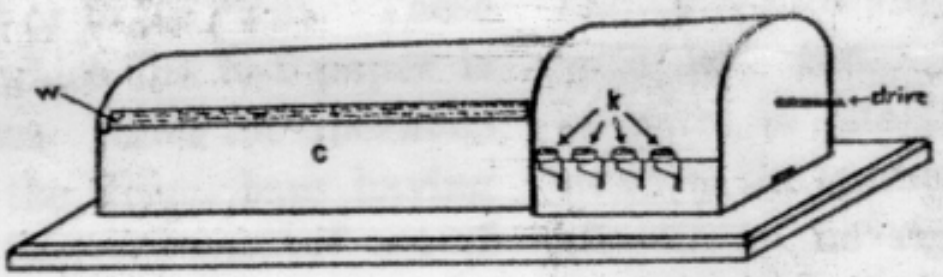


Chart I

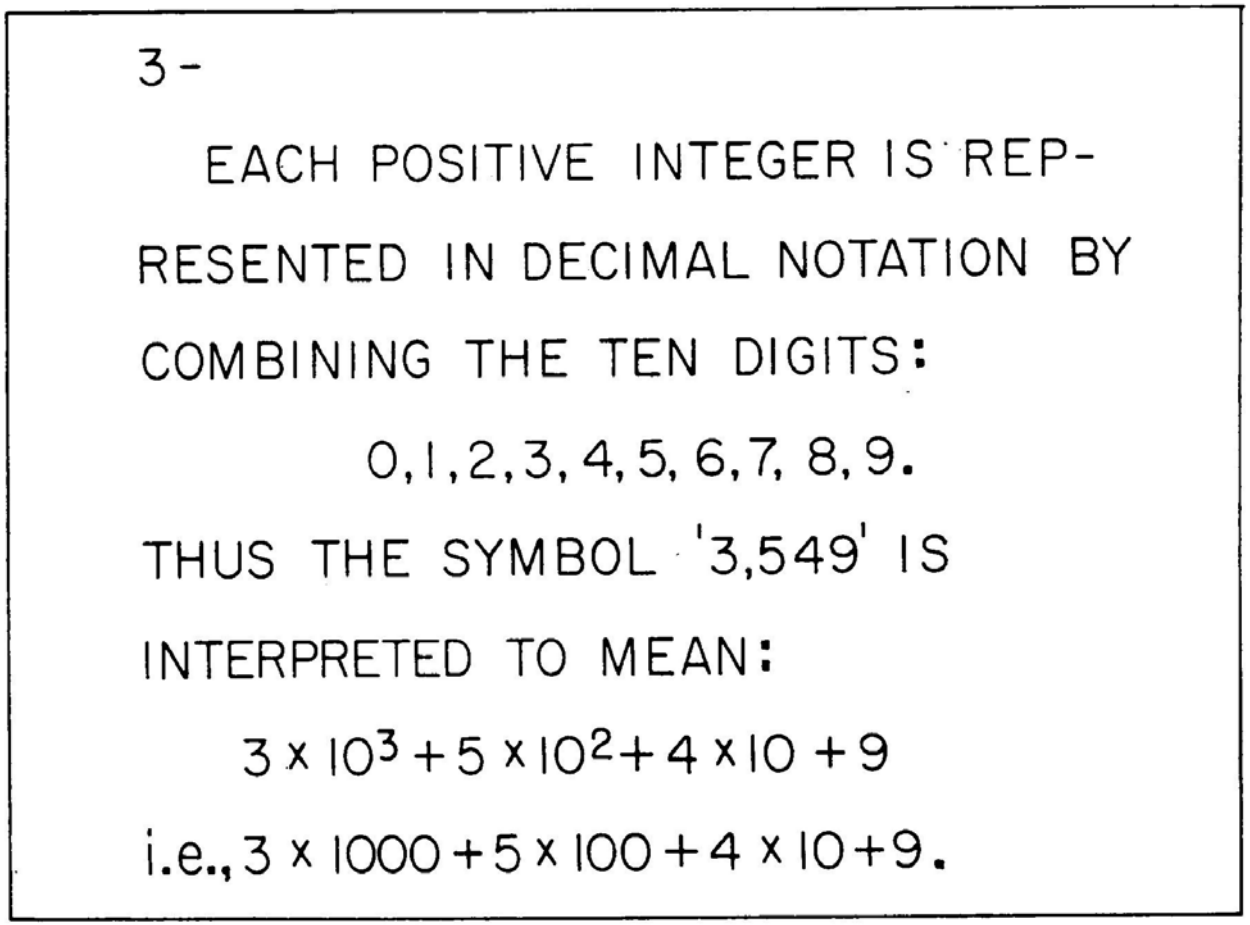
Through the window (W) in the casing there appears a question of the selective answer type, such as:

To help the poor debtors of England; James Oglethorpe founded the colony of (a) Connecticut (b) Delaware (c) Maryland (d) Georgia.

Computer-Based Tutoring Systems

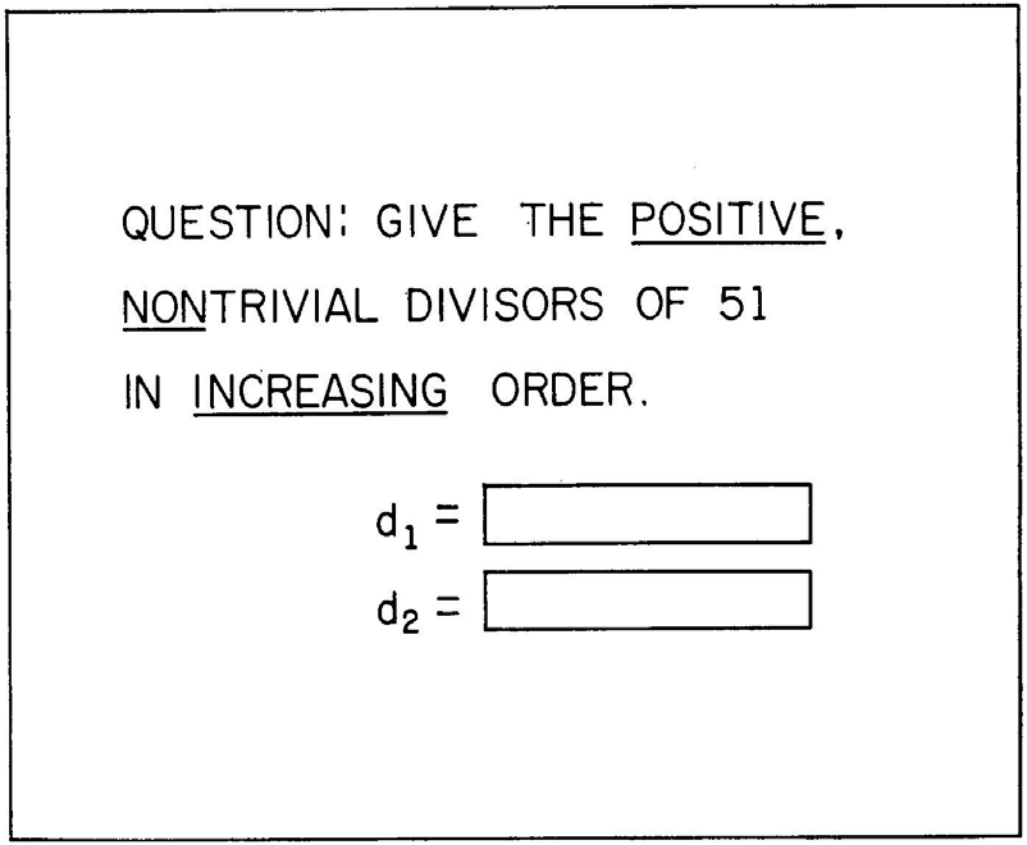
- 1961-1962
  - PLATO II
    - (Coulson, 1962)
  - CLASS
    - (Bitzer, Braunfelf, & Lichtenberger, 1962)
- One of the first reports of a computer-based instructional program to appear in the general scientific literature was published in the journal *Science* in 1969 (Suppes & Morningstar, 1969).

Example of a Frame
PLATO II, 1962



From Programmed Learning and Computer-Based Instruction (p. 210), by J.E. Coulson (Ed.), 1962, Santa Monica, CA. System Development Corporation. Copyright 1962 by System Development Corporation. Reprinted with permission of John Wiley & Sons, Inc.

Answer Slide
PLATO II



From Programmed Learning and Computer-Based Instruction (p. 210), by J.E. Coulson (Ed.), 1962, Santa Monica, CA. System Development Corporation. Copyright 1962 by System Development Corporation. Reprinted with permission of John Wiley & Sons, Inc.

High Density (HD) Overt responses to 176 frames	Low Density (LD) Overt responses to every other frame	Zero Density (ZD) Passive reading, key tapping to advance	Control for Time (CT) Passive reading, advance when HD advanced
1. A player piano is told what notes to play from a long scroll of paper with tiny holes punched through it. The paper scroll is like a script of commands that tells the piano what n---s to play.	1. A player piano is told what notes to play from a long scroll of paper with tiny holes punched through it. The paper scroll is like a script of commands that tells the piano what n---s to play.	1. A player piano is told what notes to play from a long scroll of paper with tiny holes punched through it. The paper scroll is like a script of commands that tells the piano what notes to play.	1. A player piano is told what notes to play from a long scroll of paper with tiny holes punched through it. The paper scroll is like a script of commands that tells the piano what notes to play.
2. With a player piano, the music is programmed. The scroll of paper is a script of commands that tells the player ----- what notes to play.	2. With a player piano, the music is programmed. The scroll of paper is a script of commands that tells the player piano what notes to play.	2. With a player piano, the music is programmed. The scroll of paper is a script of commands that tells the player piano what notes to play.	2. With a player piano, the music is programmed. The scroll of paper is a script of commands that tells the player piano what notes to play.
3. Like a player piano, a computer can be programmed. A computer program is like a sc---pt of commands that tells the c------r what to do.	3. Like a player piano, a computer can be programmed. A computer program is like a sc---pt of commands that tells the c------r what to do.	3. Like a player piano, a computer can be programmed. A computer program is like a script of commands that tells the computer what to do.	3. Like a player piano, a computer can be programmed. A computer program is like a script of commands that tells the computer what to do.

From “Degree of Constructed-Response Interaction in computer-Based Programmed Instruction,” by K.M. Kritch and D.E. Barstow, 1998, *Journal of Applied Behavior Analysis*, *31*, 387-398.

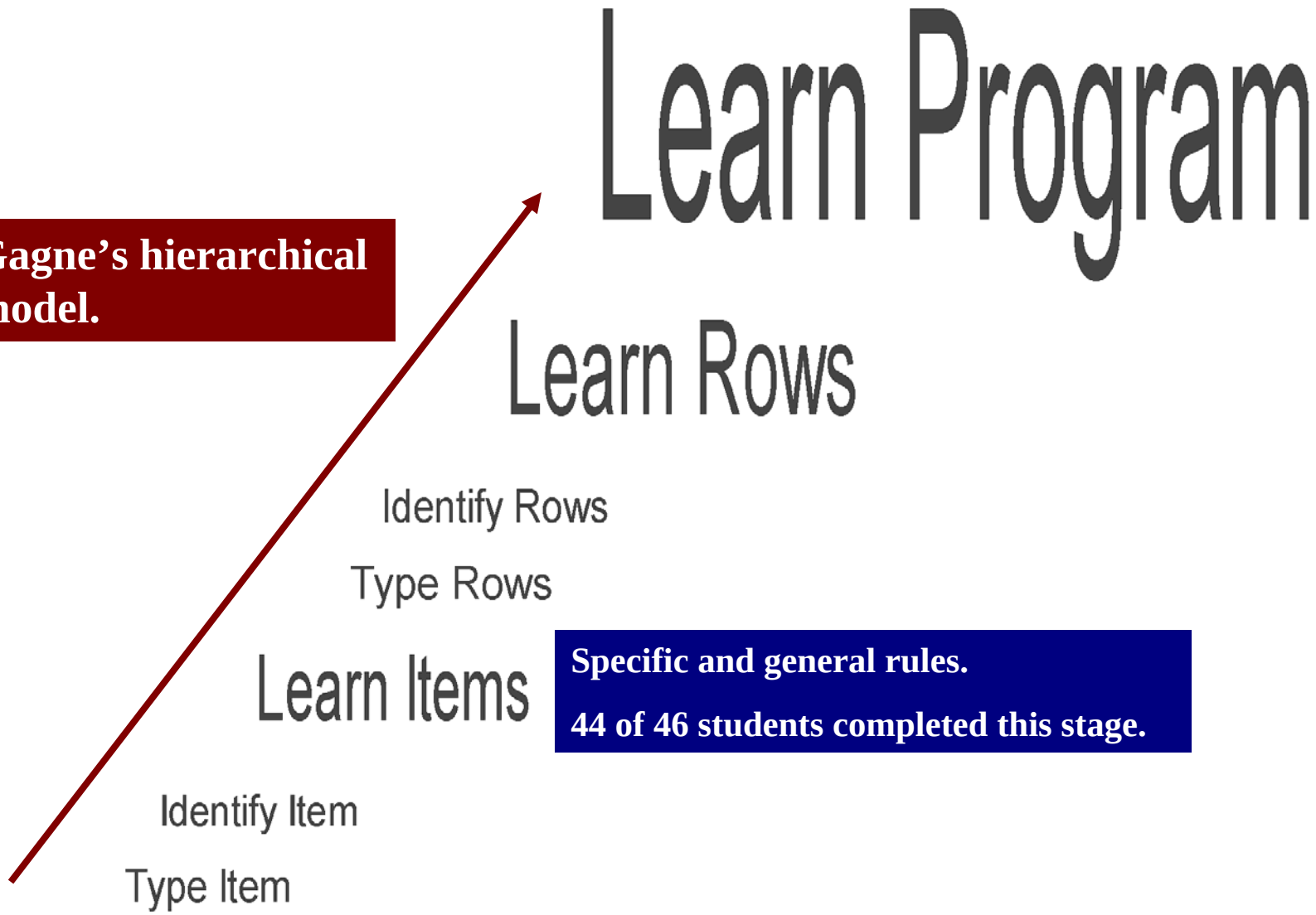
Sample Test Items
1. The "statements" that cause a computer program to take actions are called -----.
2. The command that erases any previous material from the screen is the ----- command.
3. The command that tells the program to start a new frame is the ----- ----- command.

Questionnaire Items (1=very much dislike; 2=dislike; 3=neutral; 4=like; 5=very much like)
How would you describe your "attitude" about the instructional program that you experienced today?
How would you describe your "attitude" about computer assisted instructional programs in general?
How would you describe your "attitude" about computer assisted instructional programs that specifically teach program commands like those taught in the instructional program you just experienced?

From "Degree of Constructed-Response Interaction in computer-Based Programmed Instruction," by K.M. Kritch and D.E. Barstow, 1998, *Journal of Applied Behavior Analysis*, *31*, 387-398.

Learn Unit: Greer & McDonough, 1999 Columbia University Teachers College

Gagne's hierarchical model.



Specific and general rules.

44 of 46 students completed this stage.

Preview: Spring 2002 Course Sections

- **Undergraduate class**
  - 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 1**
  - Pre-tutor questionnaires
    - Java Experience
    - Confidence in Java
  - Run the tutor
  - Post-tutor questionnaires
    - Evaluate the tutor
    - Confidence in Java
- **Class 2**
  - Run the Applet
- **Classes 3 – 14**
  - Lectures, demonstrations, supervision

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Each cell is an item to be learned.

Each row is a row to be learned.

32 items.

21 atomic units.

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

Item Learning

Item learning example: **import**

Objectives:

1. Learn the meaning of the item.
2. Learn the serial context of the item.
3. Learn the general context of the item.
4. Learn to construct the item.

import java.applet.Applet;

Serial Stream

Serial learning example:

import java.applet.Applet;

Objectives:

1. Learn the serial order of the items.

import java.applet.Applet;

Advanced Serial Stream

Serial Stream with Advanced Units

Objectives:

1. Learn the serial context of a group of items.
2. Learn the general context of a group of items.
3. Learn to construct a group of items as a single unit.

```
import java.applet.Applet;
import java.awt.Label;
```

UMBC
AN HONORS UNIVERSITY IN MARYLAND

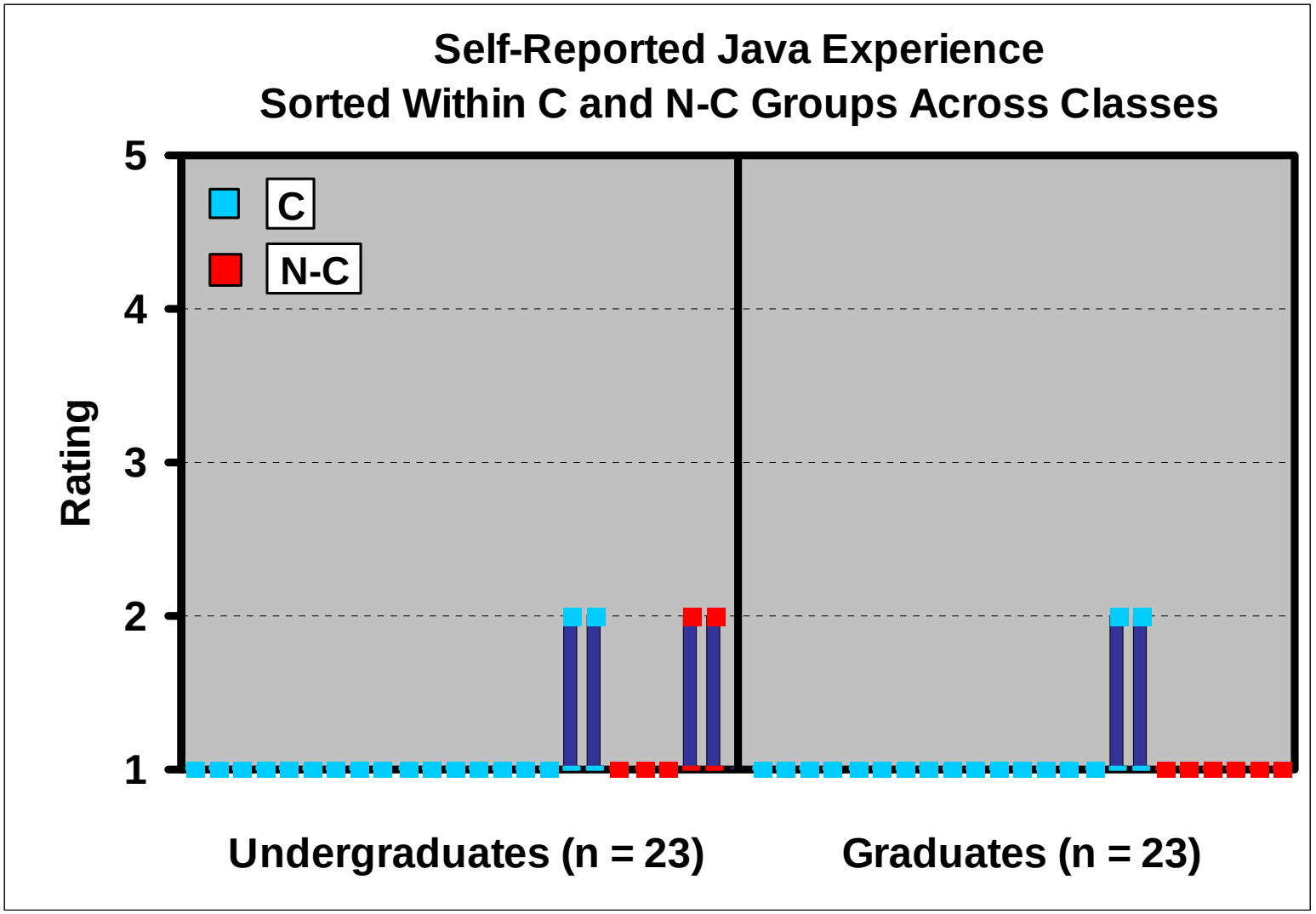
Serial Stream as a Unit (Intraverbal-Plus)

Serial Stream Unit

Objectives:

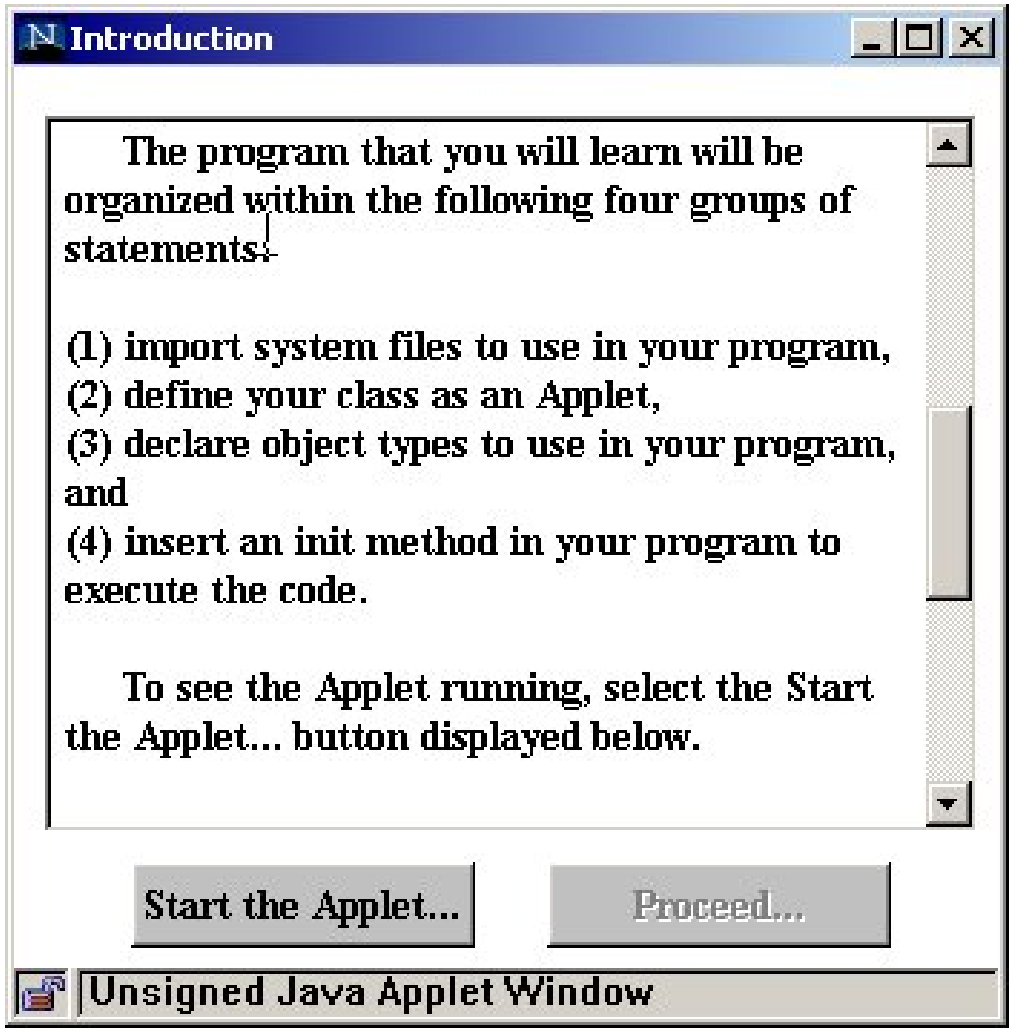
1. Learn the serial stream as one unit.

```
import java.applet.Applet;
import java.awt.Label;
public class myProgram extends Applet {
Label myLabel;
public void init() {
...
}
```

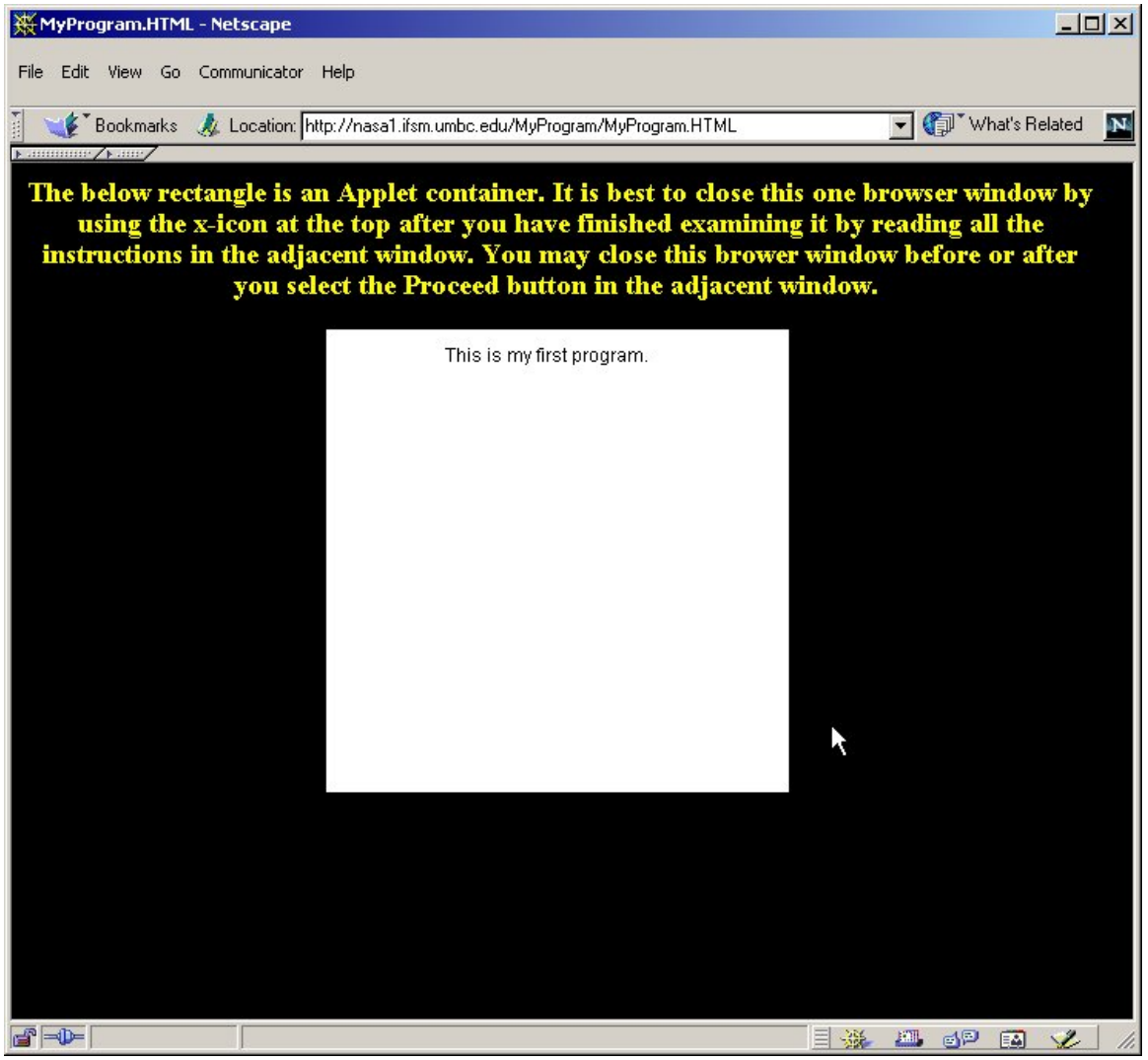


1 = No experience. (I am a novice in Java.)
5 = Extensive experience. (I am an expert in Java.)

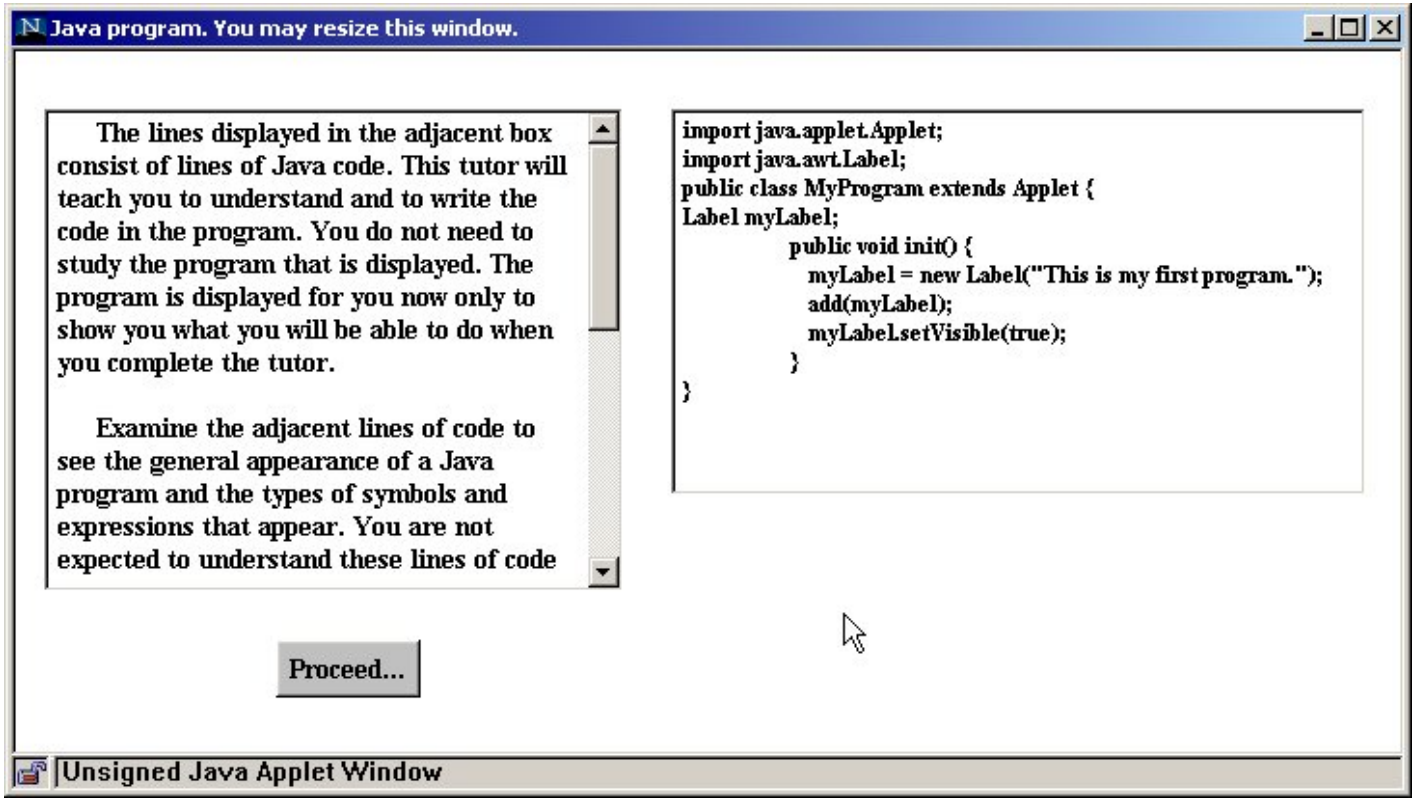
Advanced Organizers



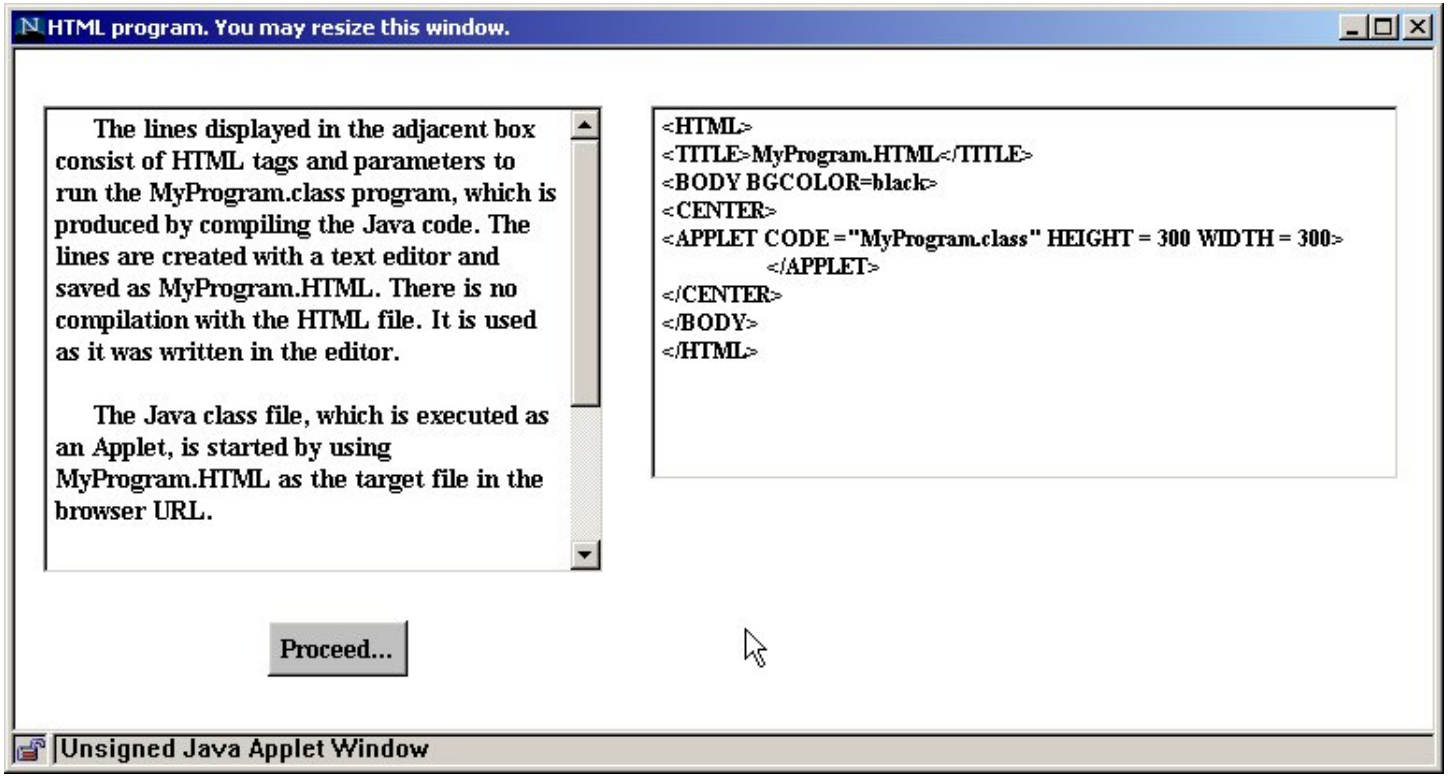
Run the Applet



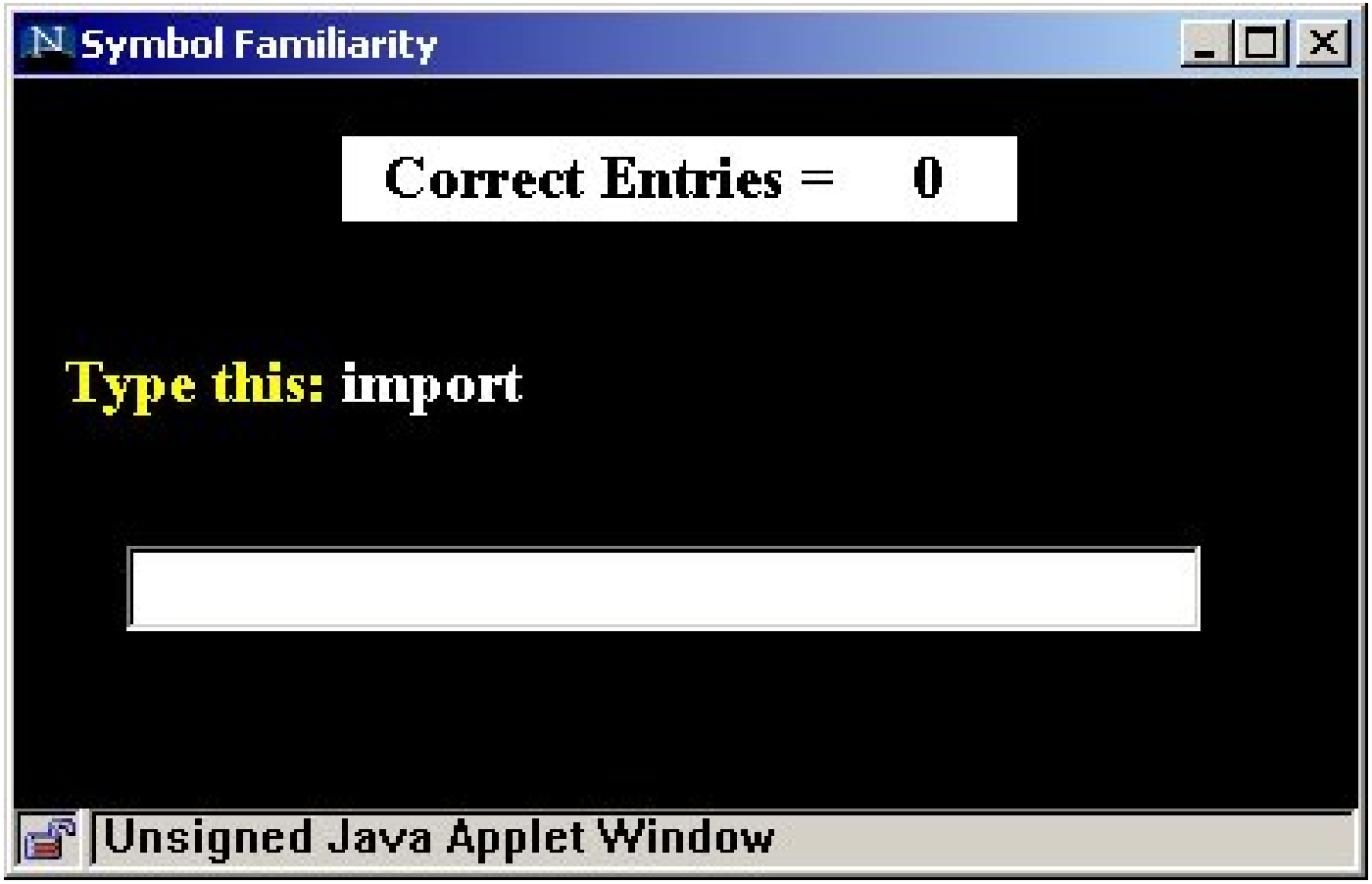
Program Overview



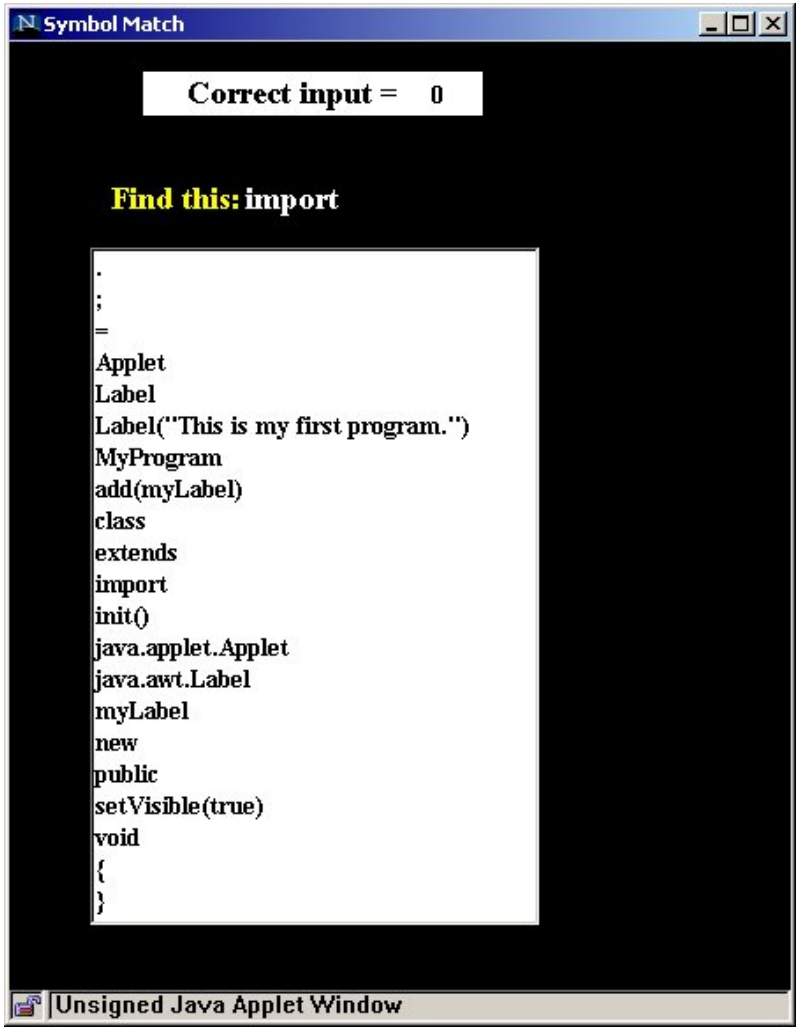
HTML Overview



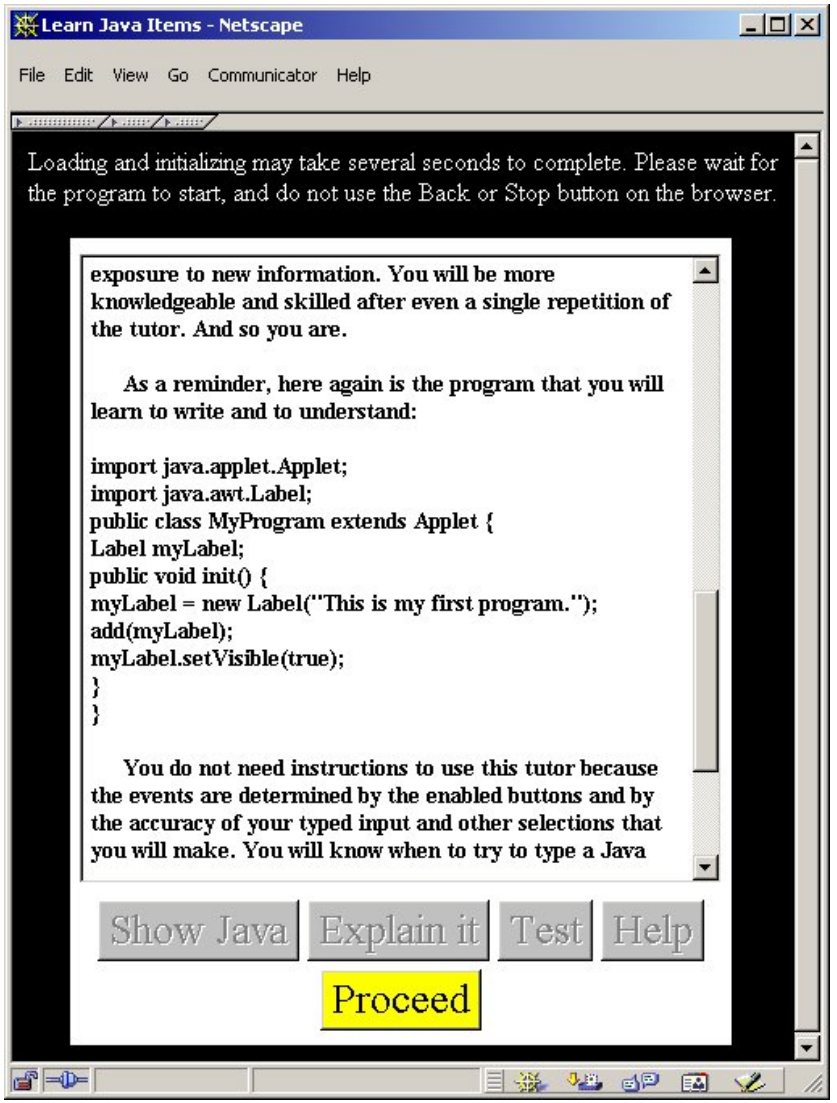
Item Familiarity Interface (Input Errors)



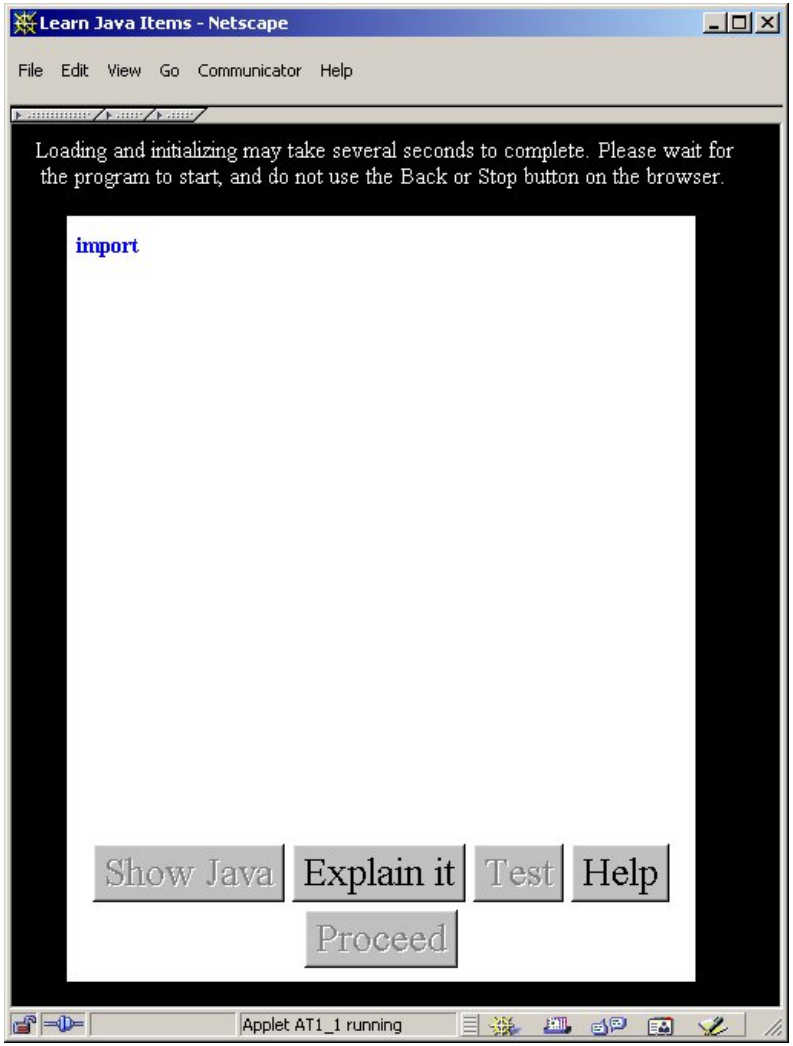
Item Identification (Selection Errors)



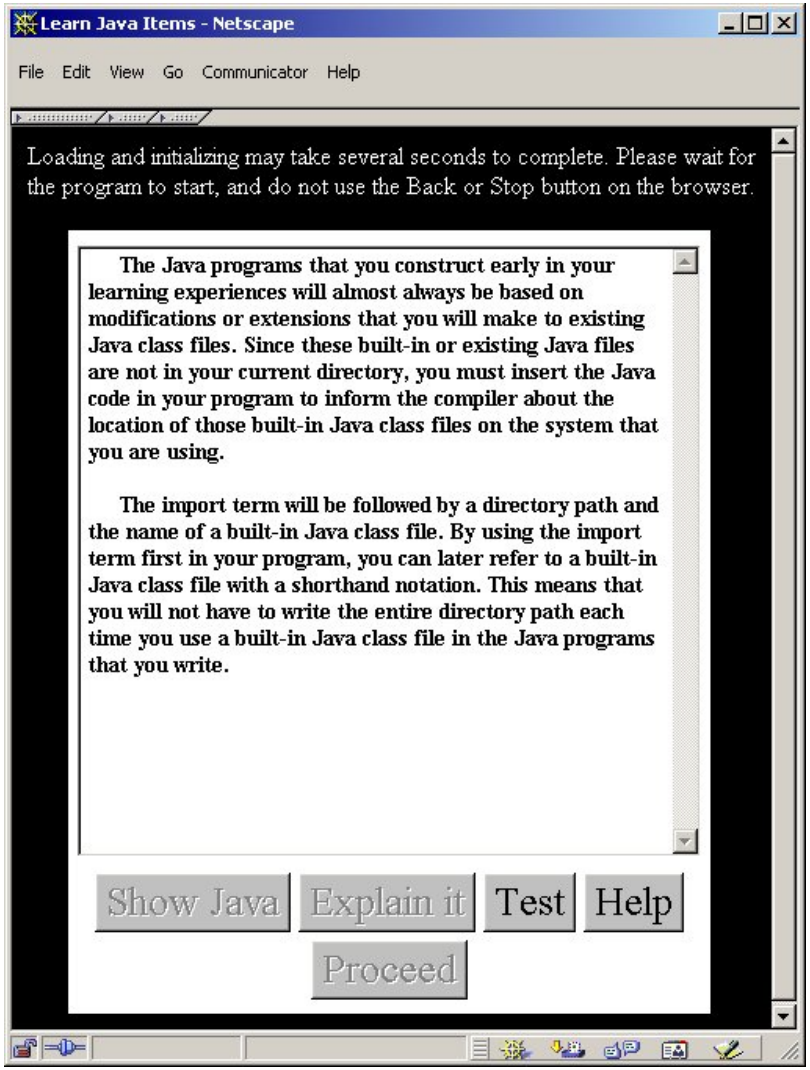
Item Learning



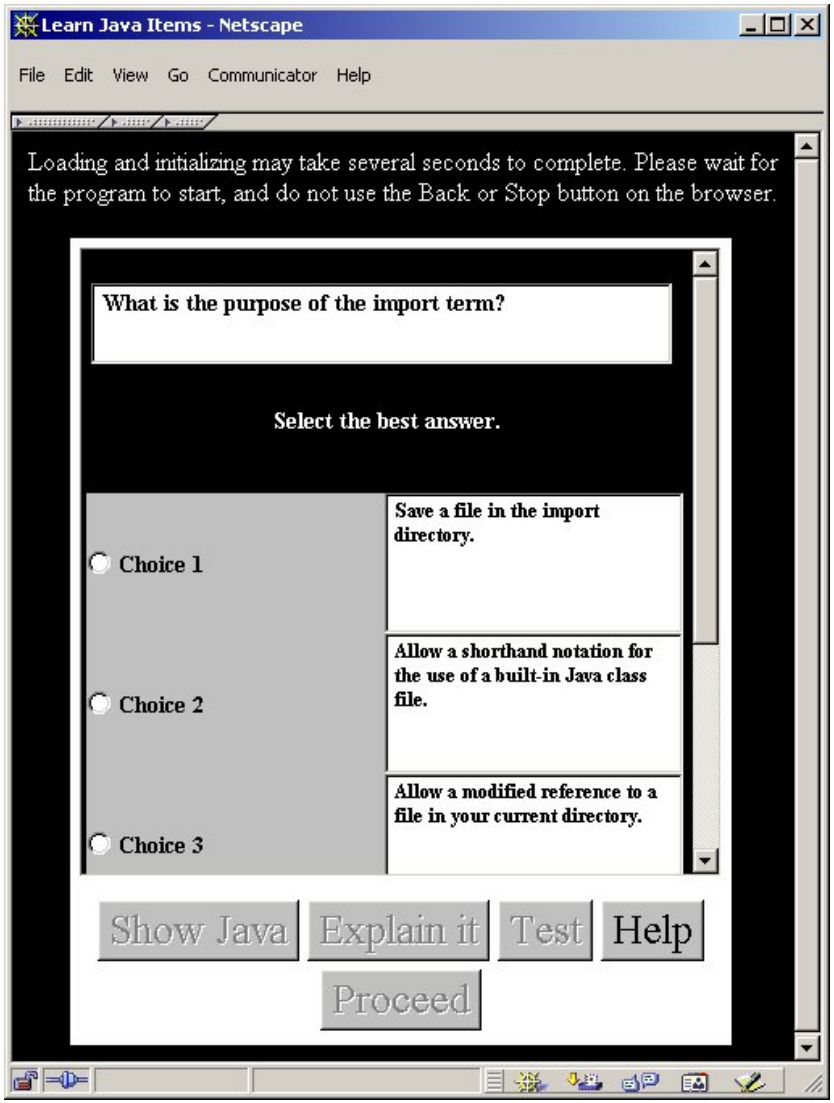
Observe the Item in Context



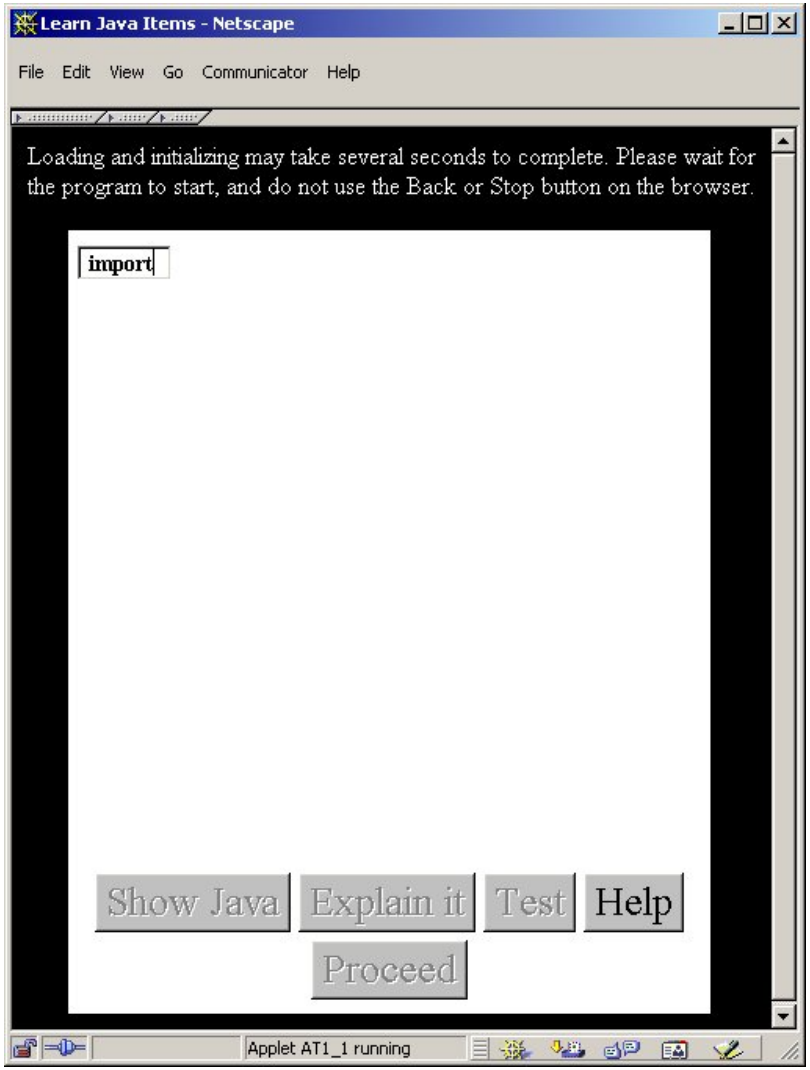
Read the Description (General Rules)



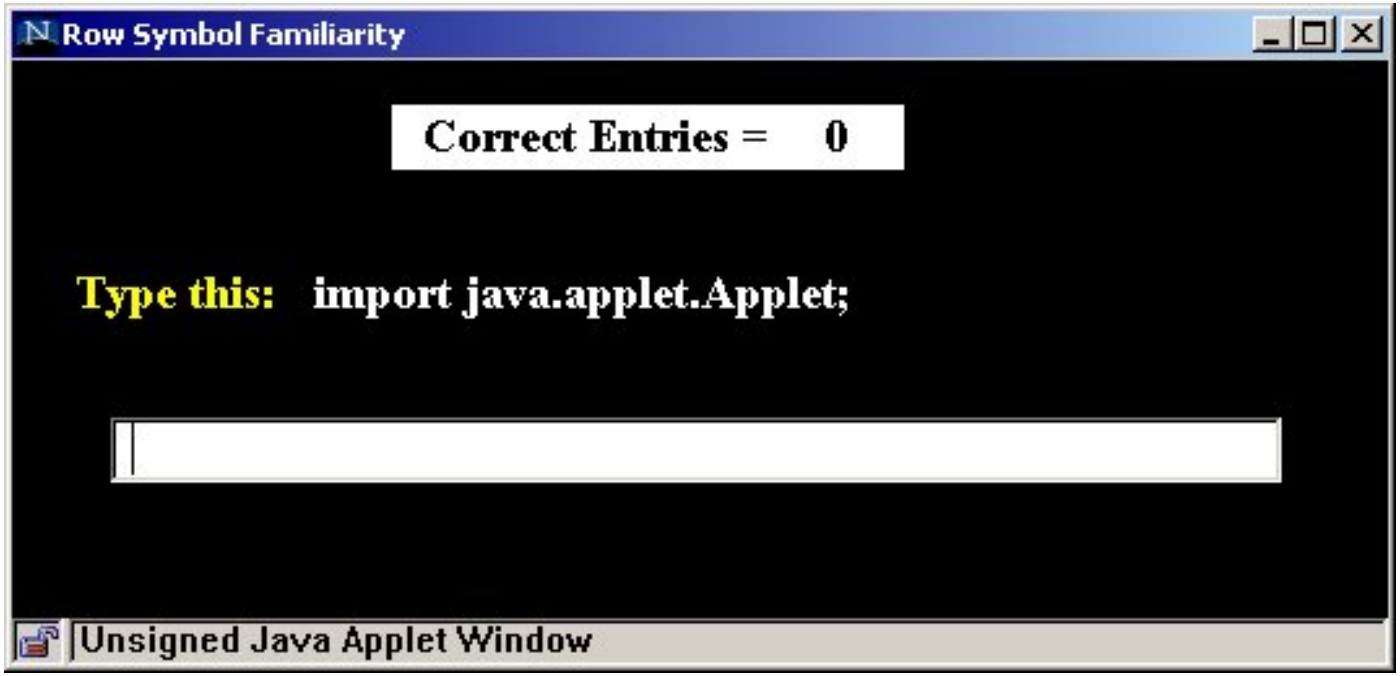
Item Multiple-Choice Test (Selection Errors)



Input the Item from Recall (Input Errors)

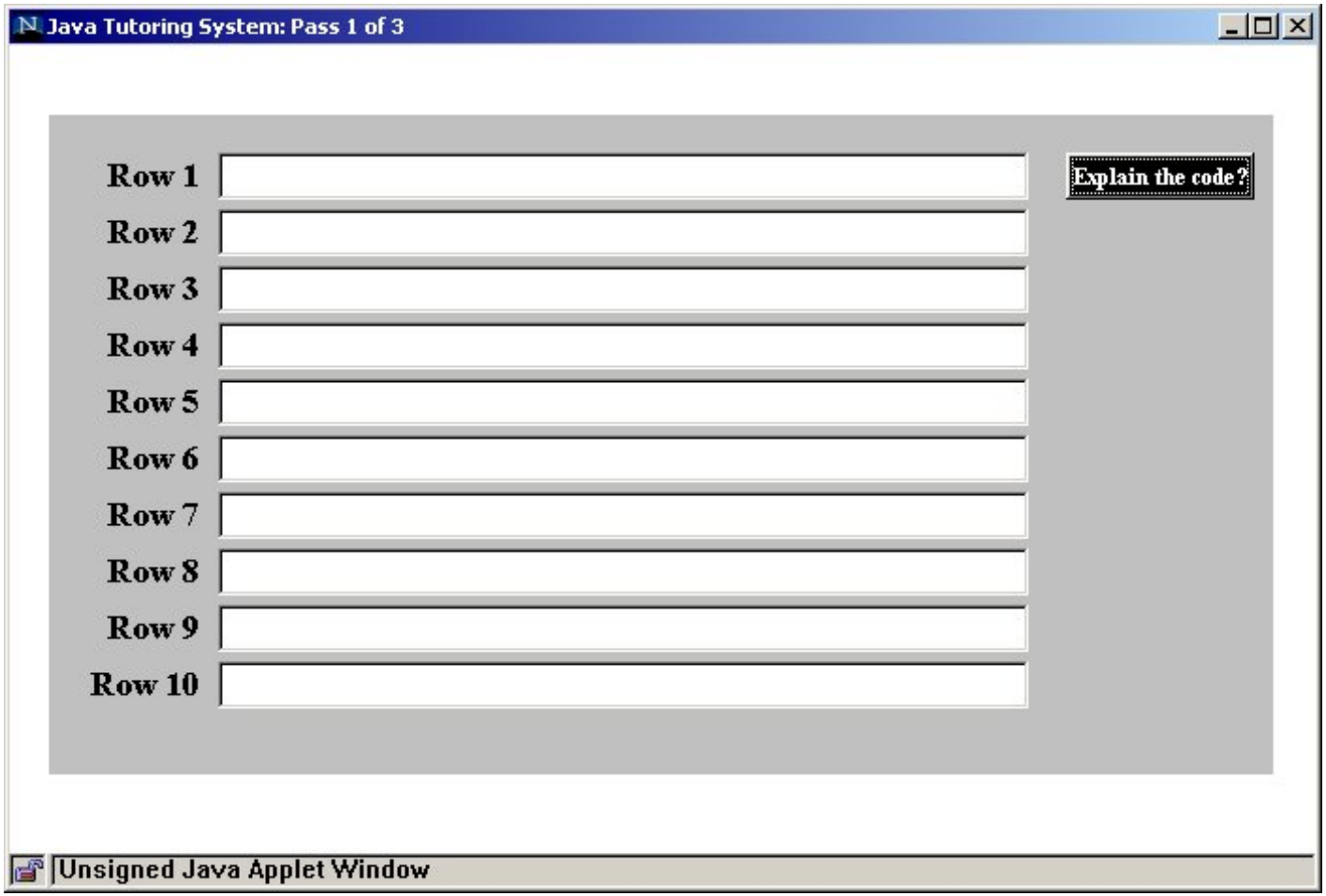


Row Familiarity (Input Errors)



Row Identification (Selection Errors)

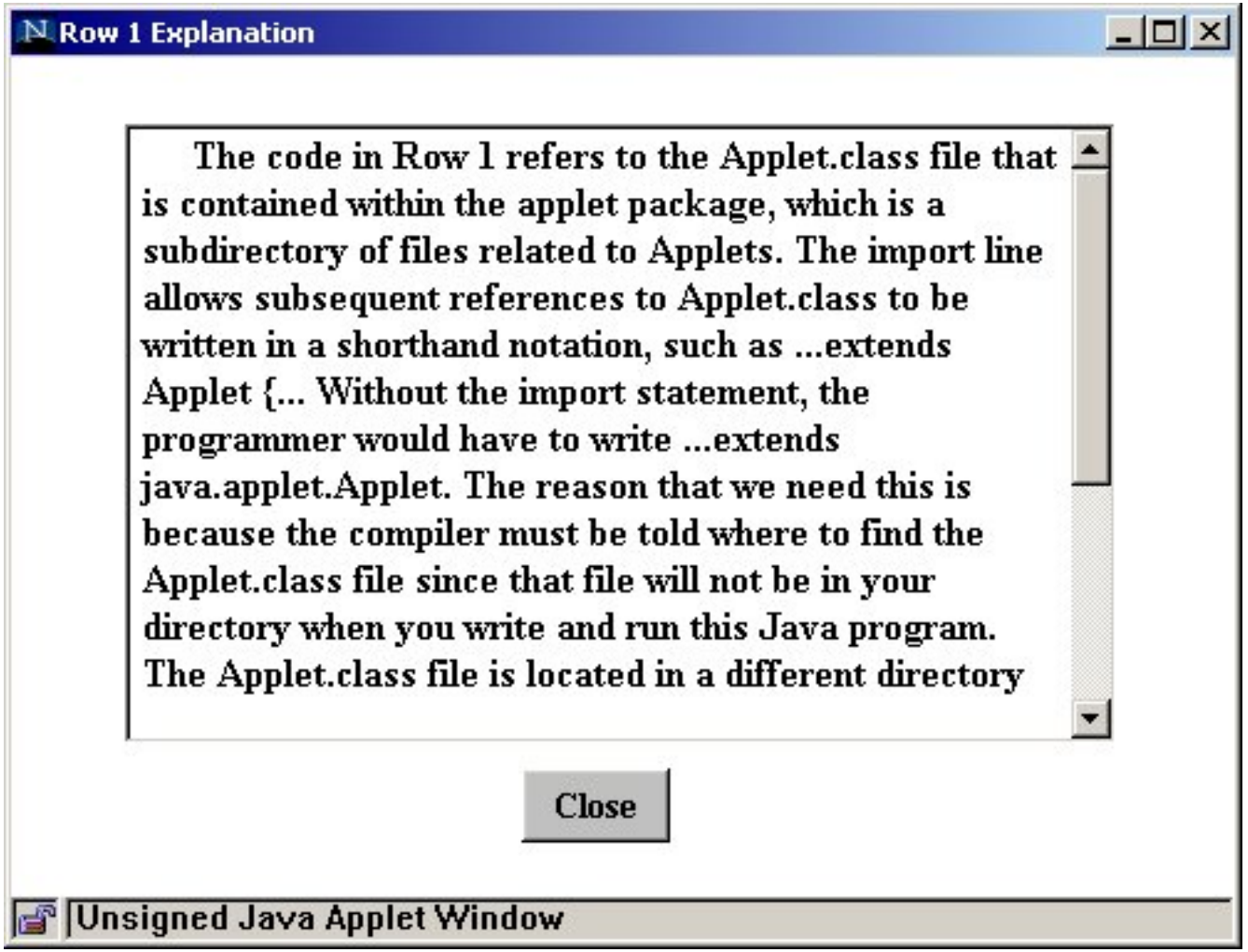
Row Interface (Input and Test Errors)



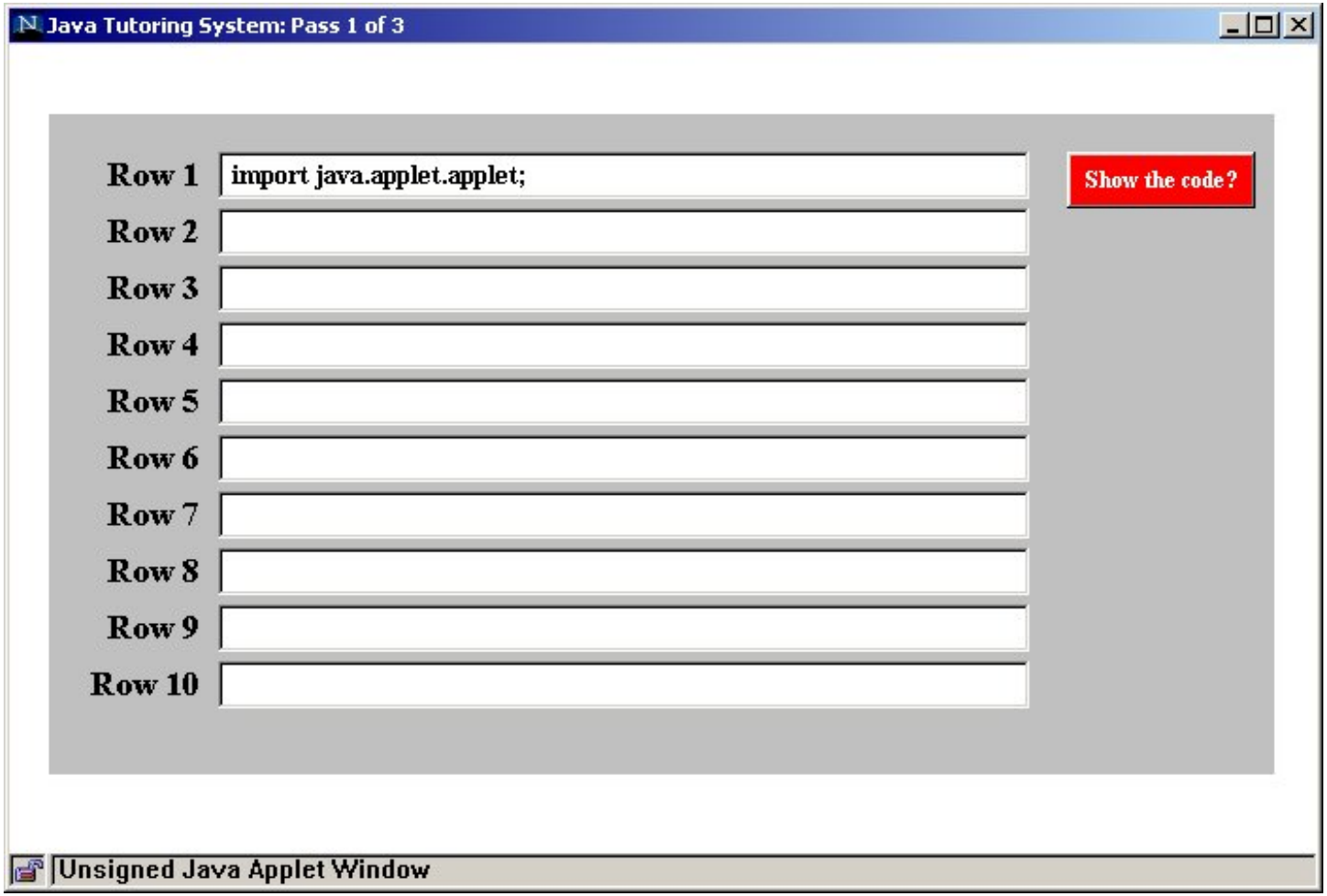
Row Interfaces

- **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

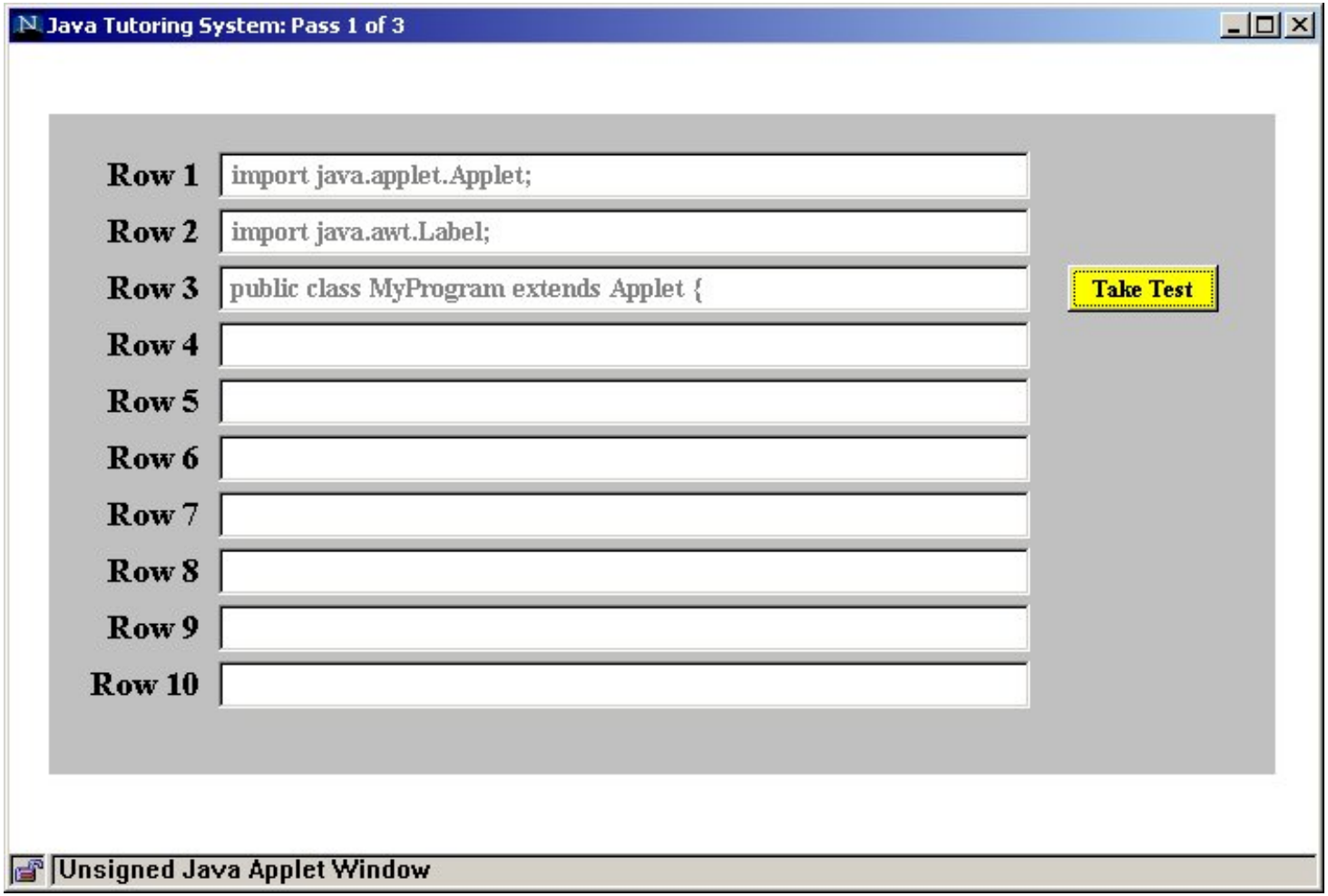
Explanation of a Row



Observe the Code



Take Test After Correct Input

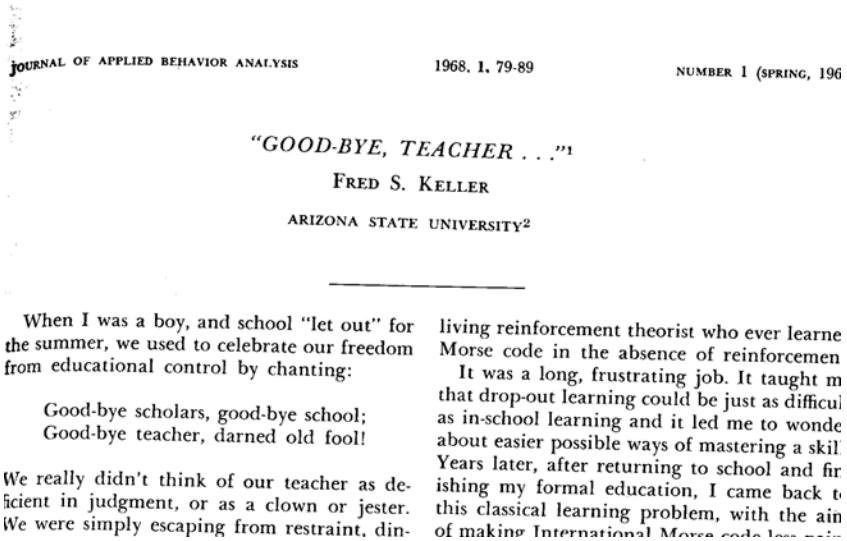


Text Window (Input Errors)

Next Class Period

1. Modified Personalized System of Instruction

   Lecture and Collaborate



JOURNAL OF APPLIED BEHAVIOR ANALYSIS 1968, 1, 79-89 NUMBER 1 (SPRING, 196

"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, din-

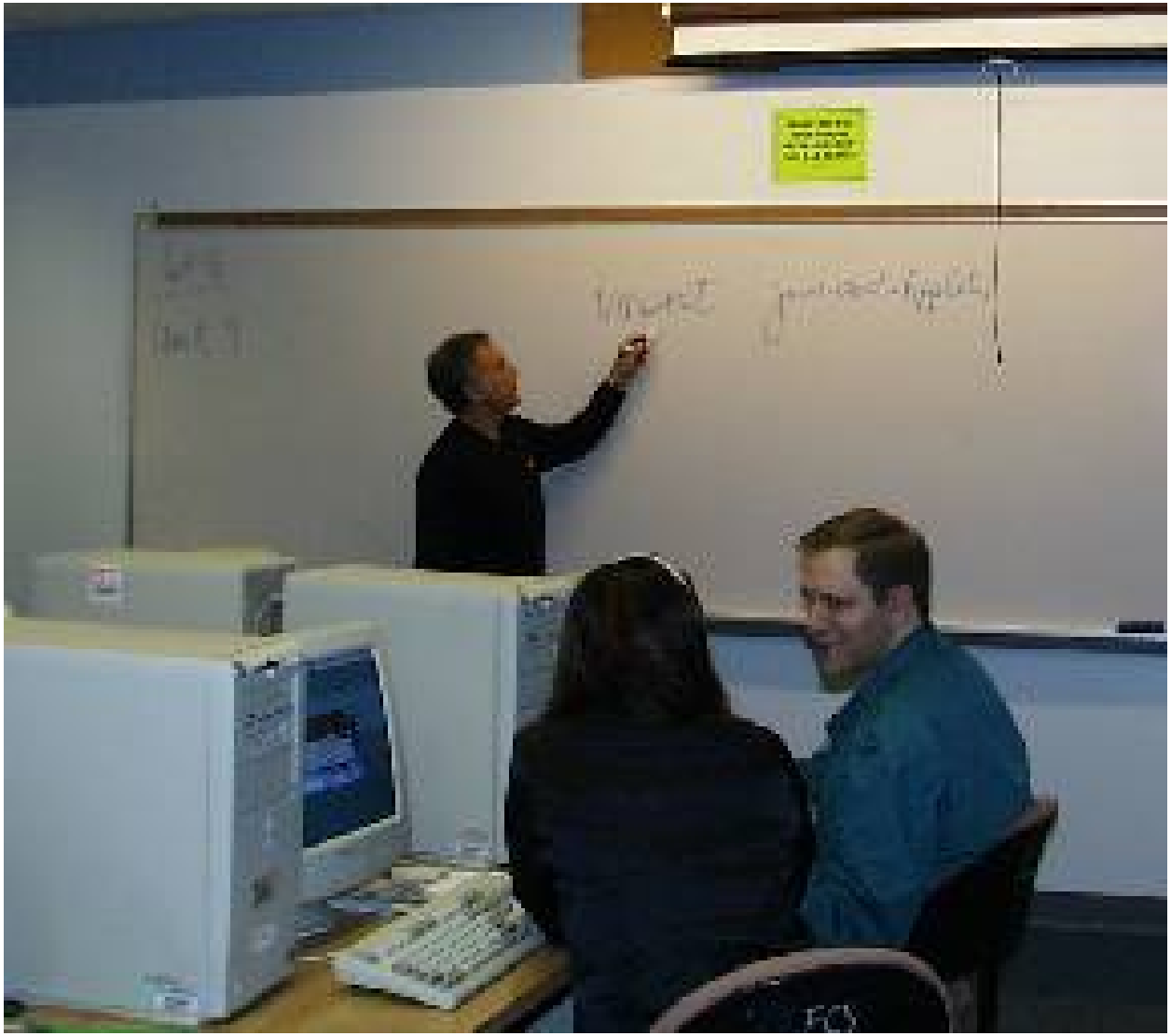
living reinforcement theorist who ever learne Morse code in the absence of reinforcemen

It was a long, frustrating job. It taught m that drop-out learning could be just as difficul as in-school learning and it led me to wonde about easier possible ways of mastering a skil Years later, after returning to school and fir ishing my formal education, I came back t this classical learning problem, with the ain of making International Morse code less pain-

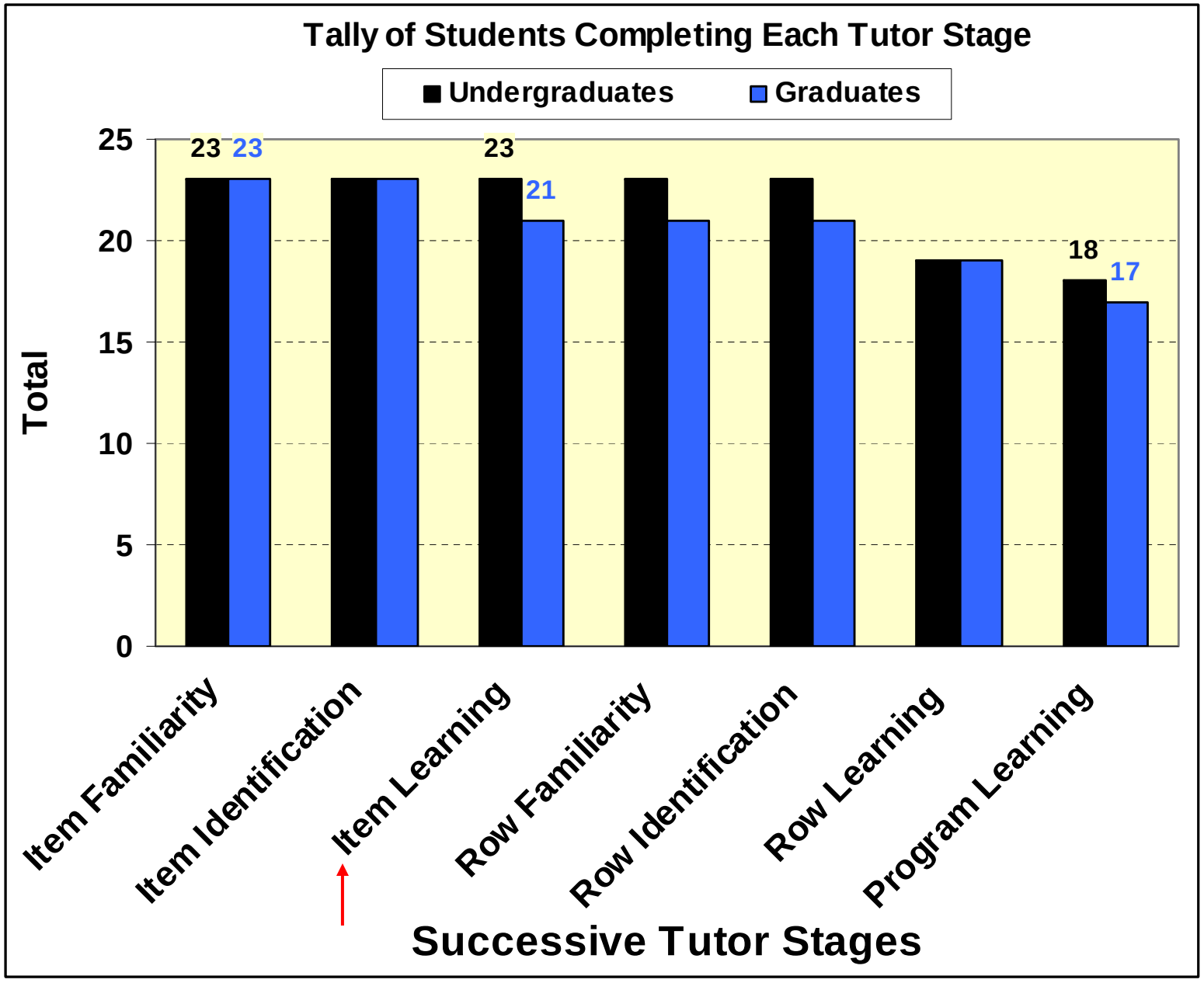
2. Run the Applet

UMBC
AN HONORS UNIVERSITY IN MARYLAND

UMBC
AN HONORS UNIVERSITY IN MARYLAND



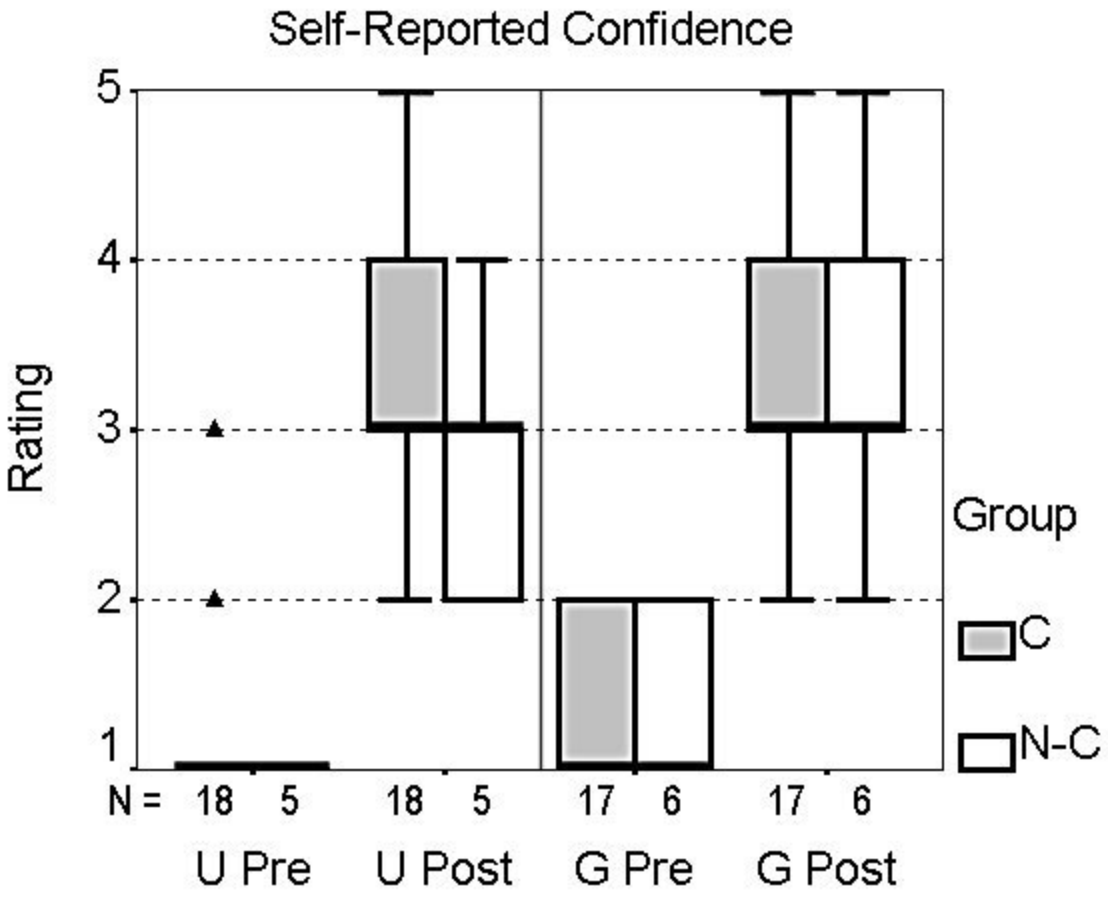
UMBC
AN HONORS UNIVERSITY IN MARYLAND


Tally of Students Completing Each Tutor Stage
Undergraduates
Graduates
Total
25
20
15
10
5
0
23
23
23
21
18
17
Item Familiarity
Item Identification
Item Learning
Row Familiarity
Row Identification
Row Learning
Program Learning
Successive Tutor Stages

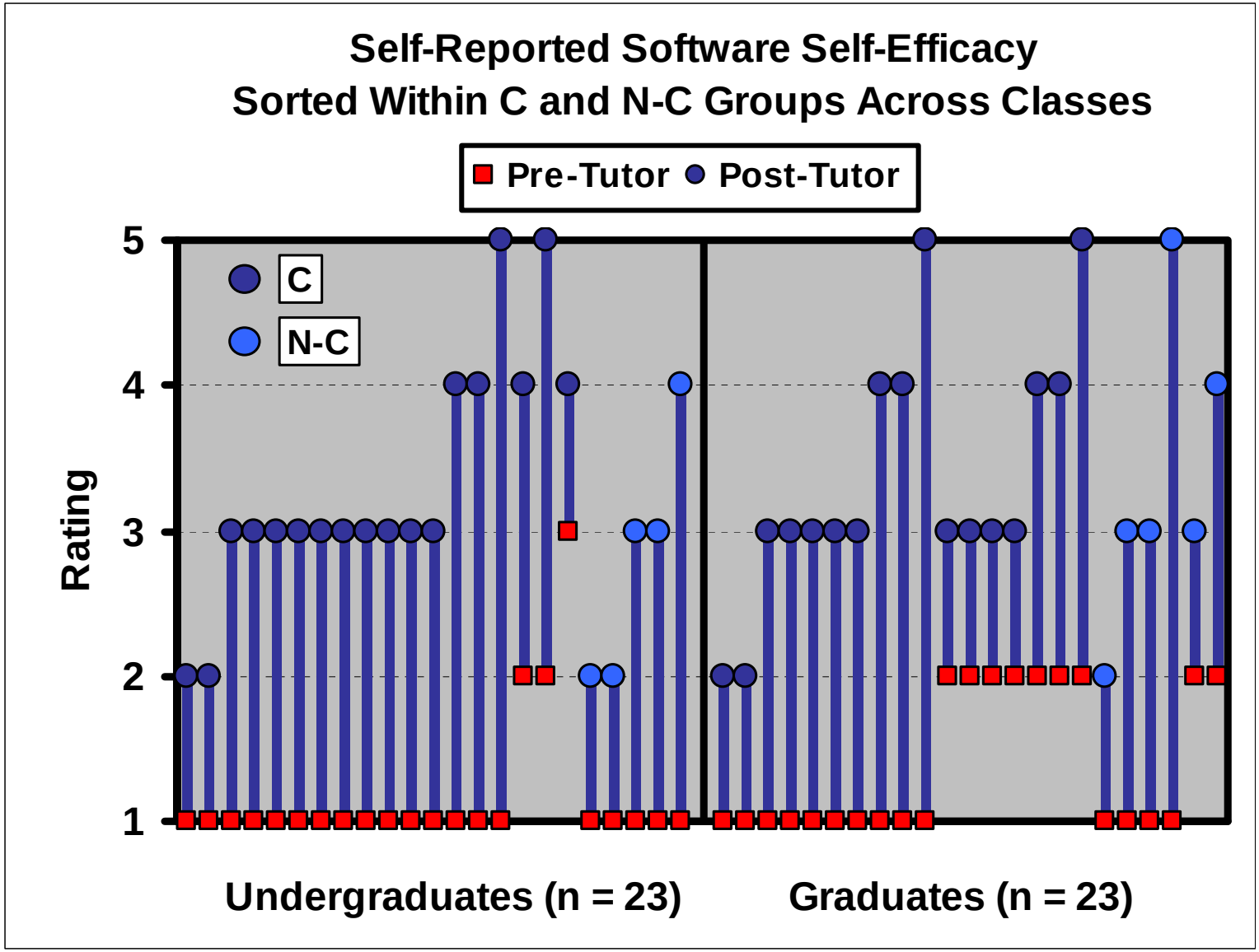
Software Self-Efficacy: Confidence in Using the Items of Code

Pre-tutor alpha = .94.

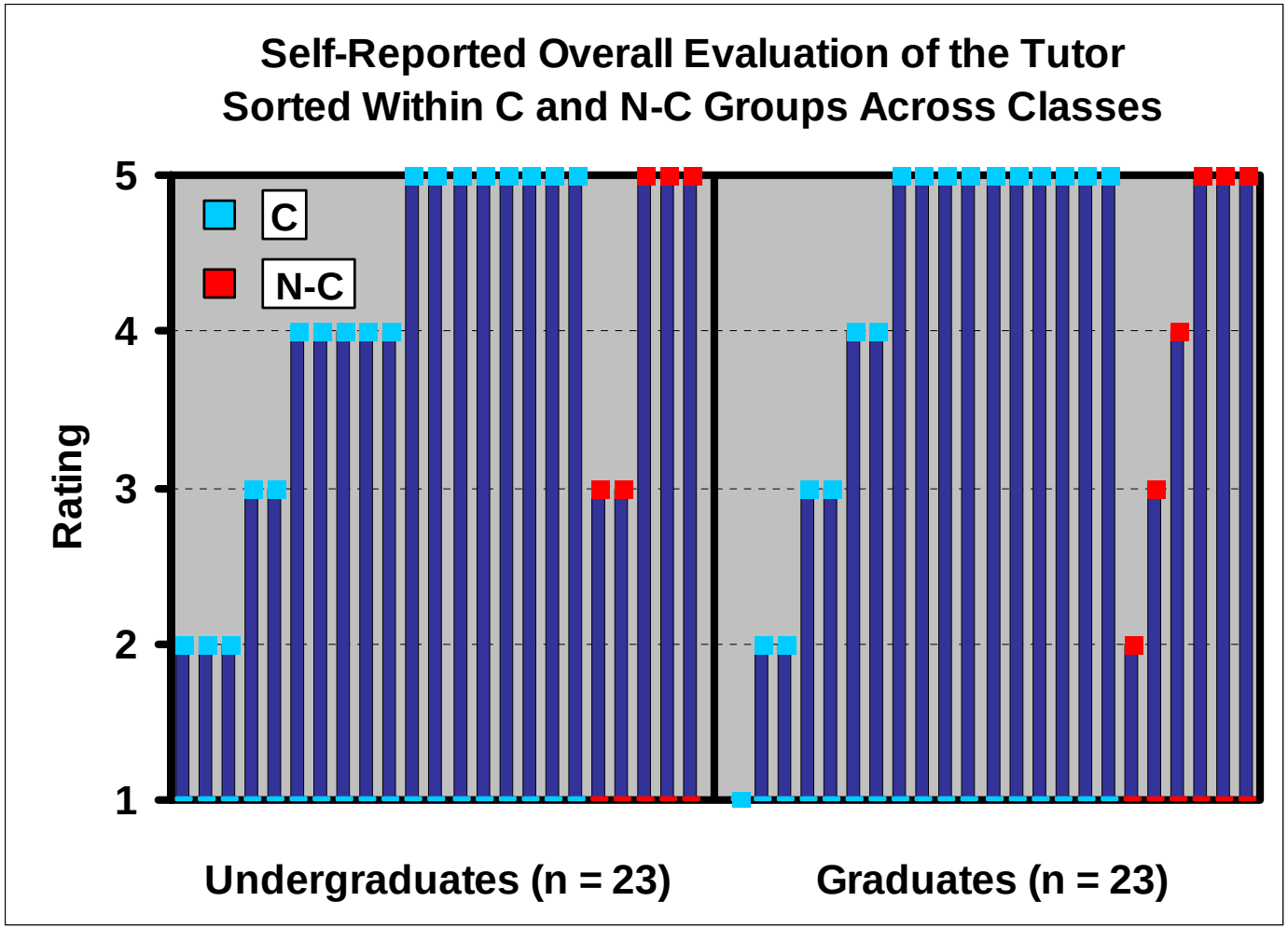
Post-tutor alpha = .90.



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



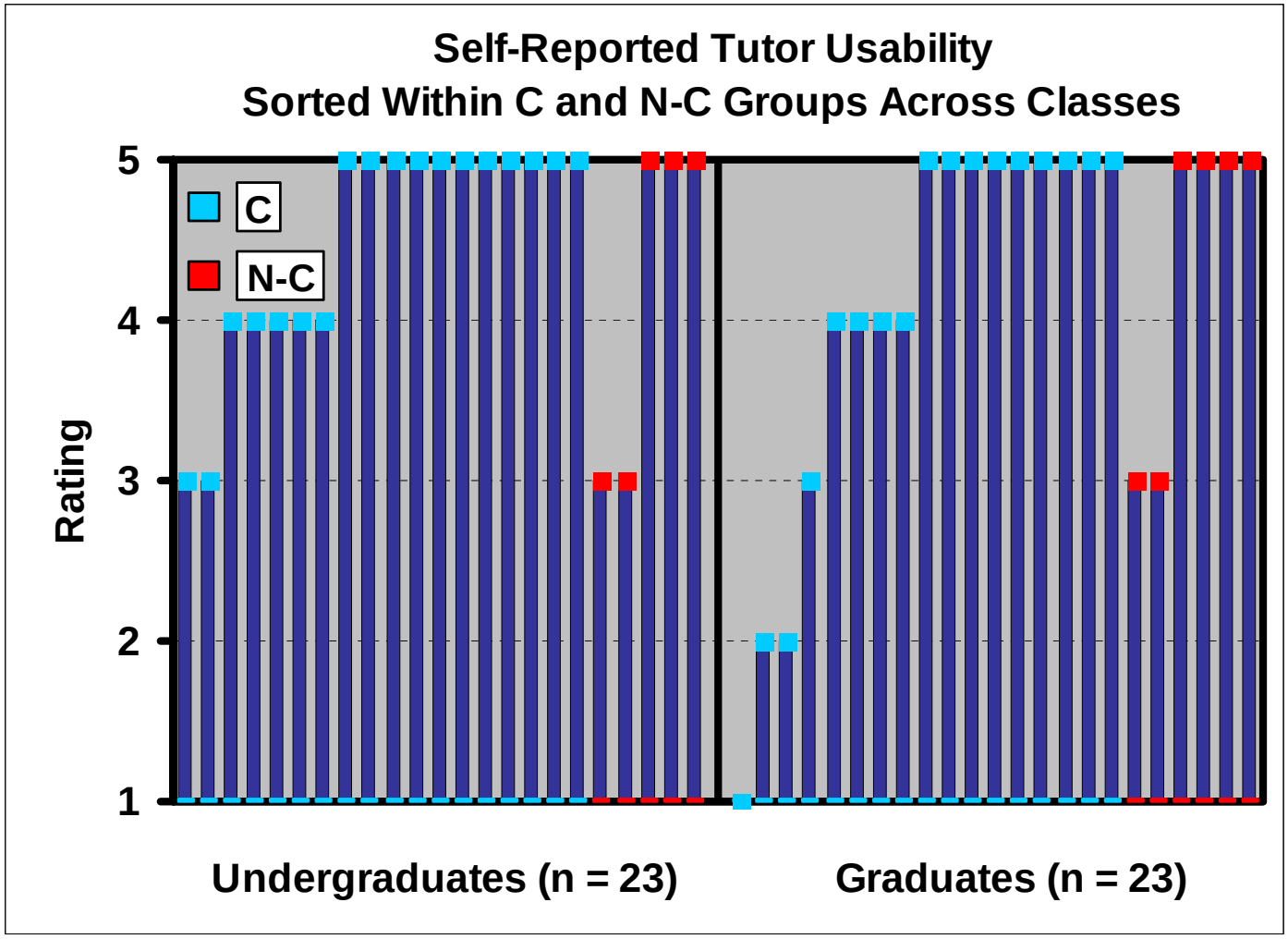
1 = Not at all confident. I do not know how to use the symbol.
5 = Totally confident. I know how to use the symbol.



1 = Totally negative. I did not like the tutor.
5 = Totally positive. I liked the tutor.

U vs. G: Kruskal-Wallis Chi-Square = 0.07, p > .05.

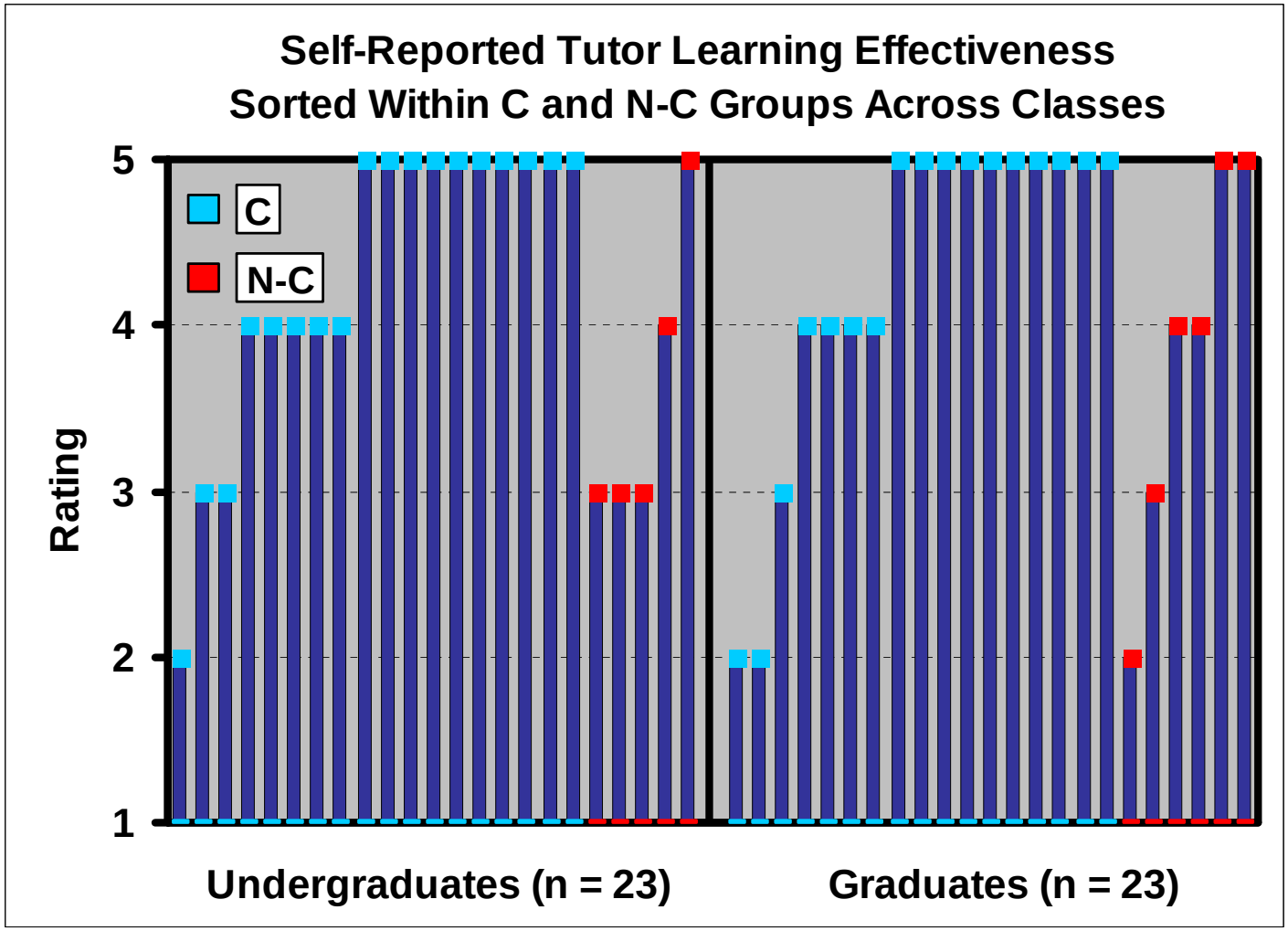
C vs. N-C: Kruskal-Wallis Chi-Square = 0.02, p > .05.



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

U vs. G: Kruskal-Wallis Chi-Square = 0.37, p > .05.

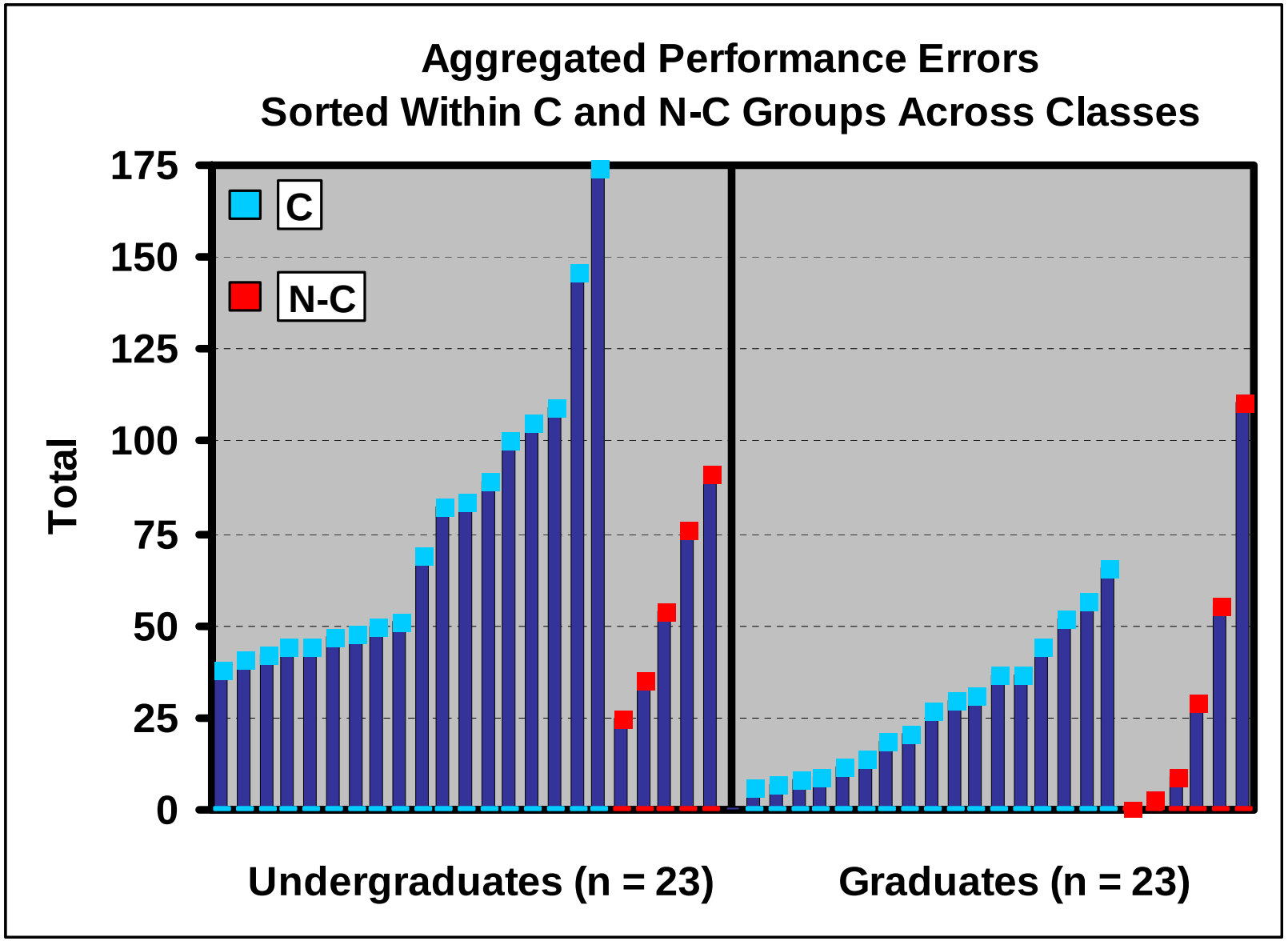
C vs. N-C: Kruskal-Wallis Chi-Square = 0.00, p > .05.



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

U vs. G: Kruskal-Wallis Chi-Square = 0.04, p > .05.

C vs. N-C: Kruskal-Wallis Chi-Square = 3.44, p > .05.

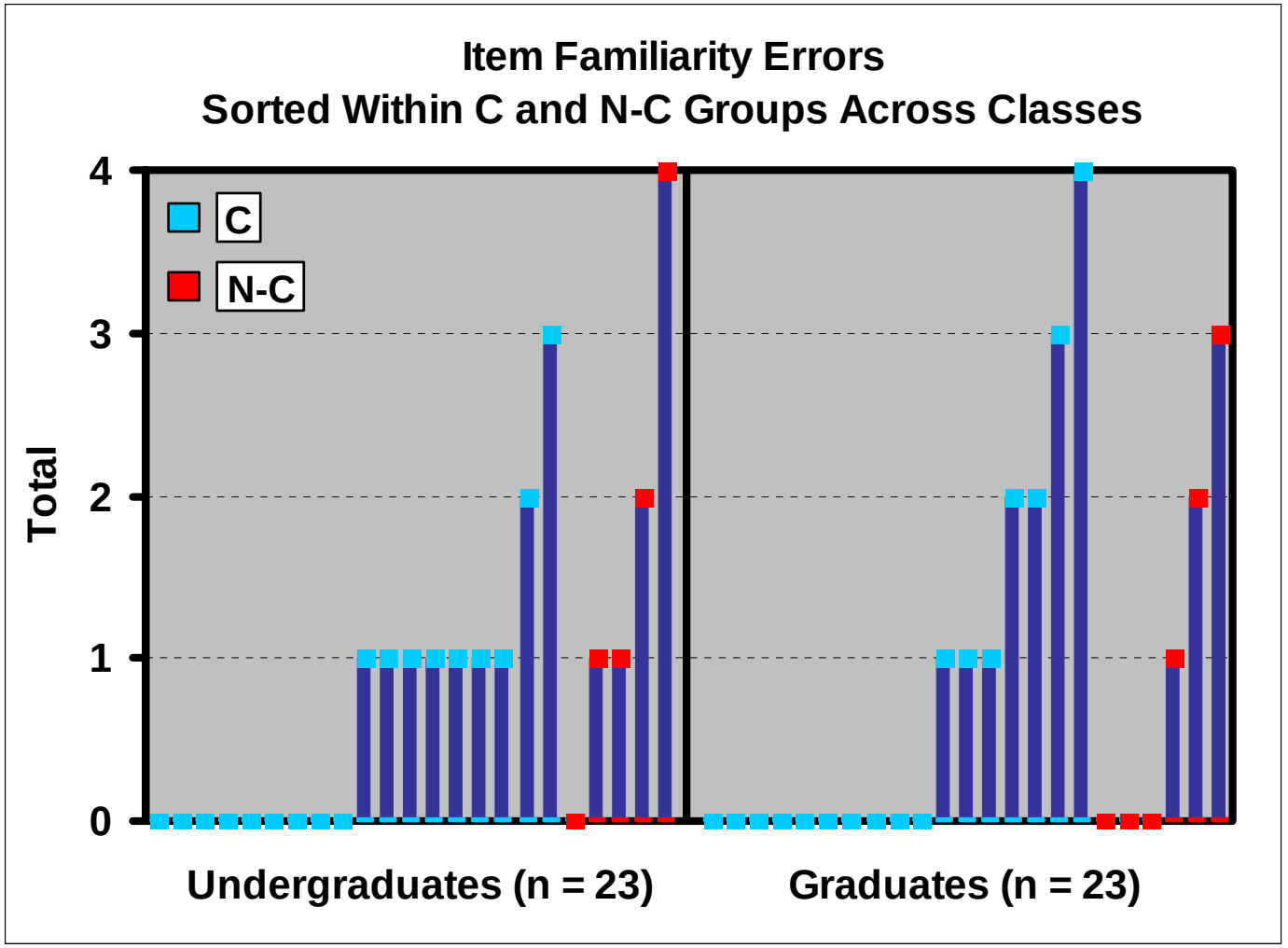


U vs. G: Kruskal-Wallis Chi-Square = 16.23, p < .01.

Binomial Test: C and N-C Groups

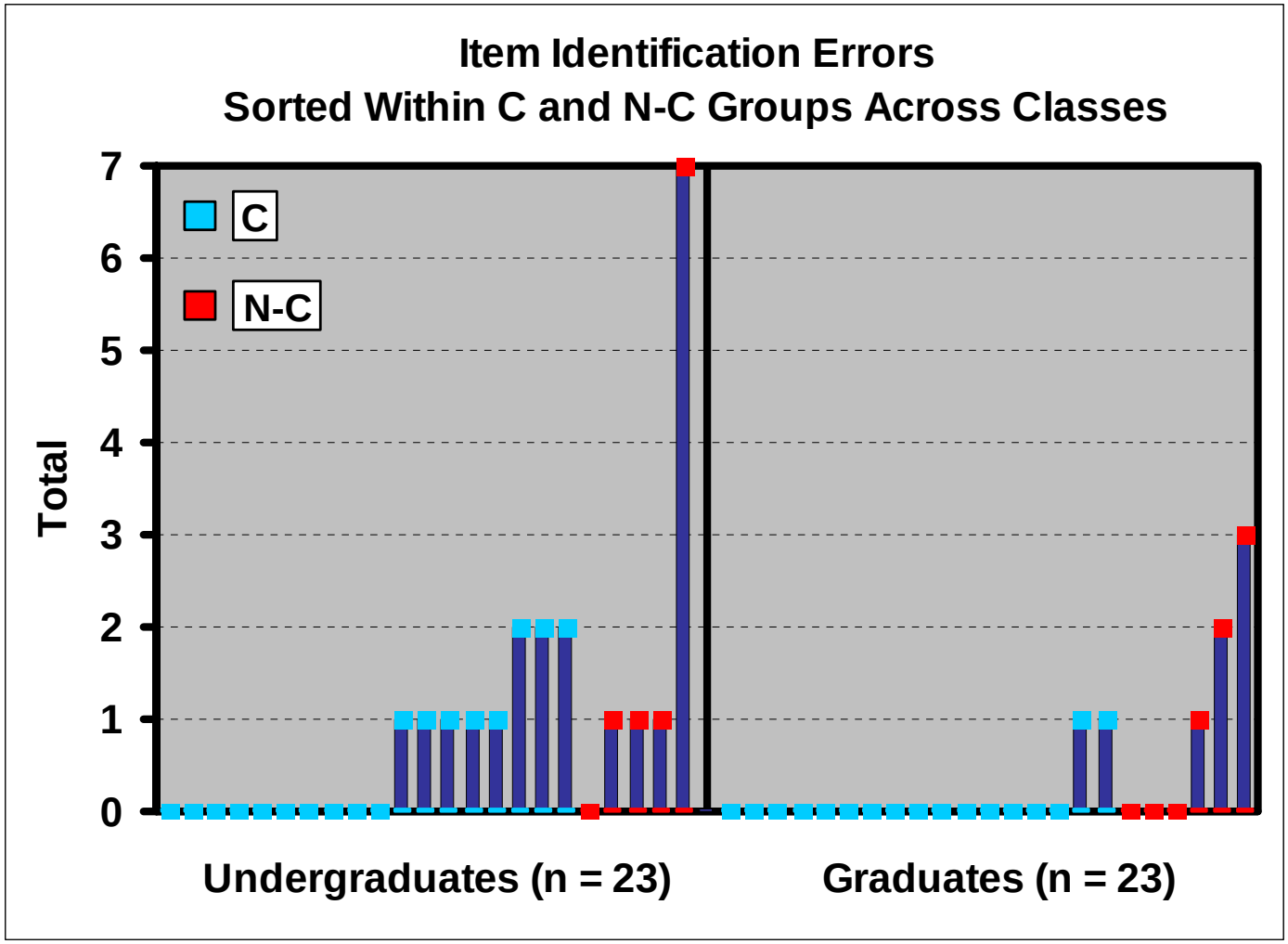
- For a final “multivariate” comparison between completers and non-completers, a binomial test was conducted based on the **item familiarity, item identification, item, and item test interfaces**. Too few non-completers were represented in the remaining interfaces for a meaningful comparison, even though the test requires no assumptions about distribution or sample size.
- Within each class, a “+” was assigned if the mean for the non-completer group was higher than the mean for the corresponding completer group. If the mean was lower, a “-“ was assigned. **For the eight comparisons, there were eight “+” outcomes**.
- A binomial test showed that the probability of this outcome occurring by chance is **0.004**.

UMBC
AN HONORS UNIVERSITY IN MARYLAND



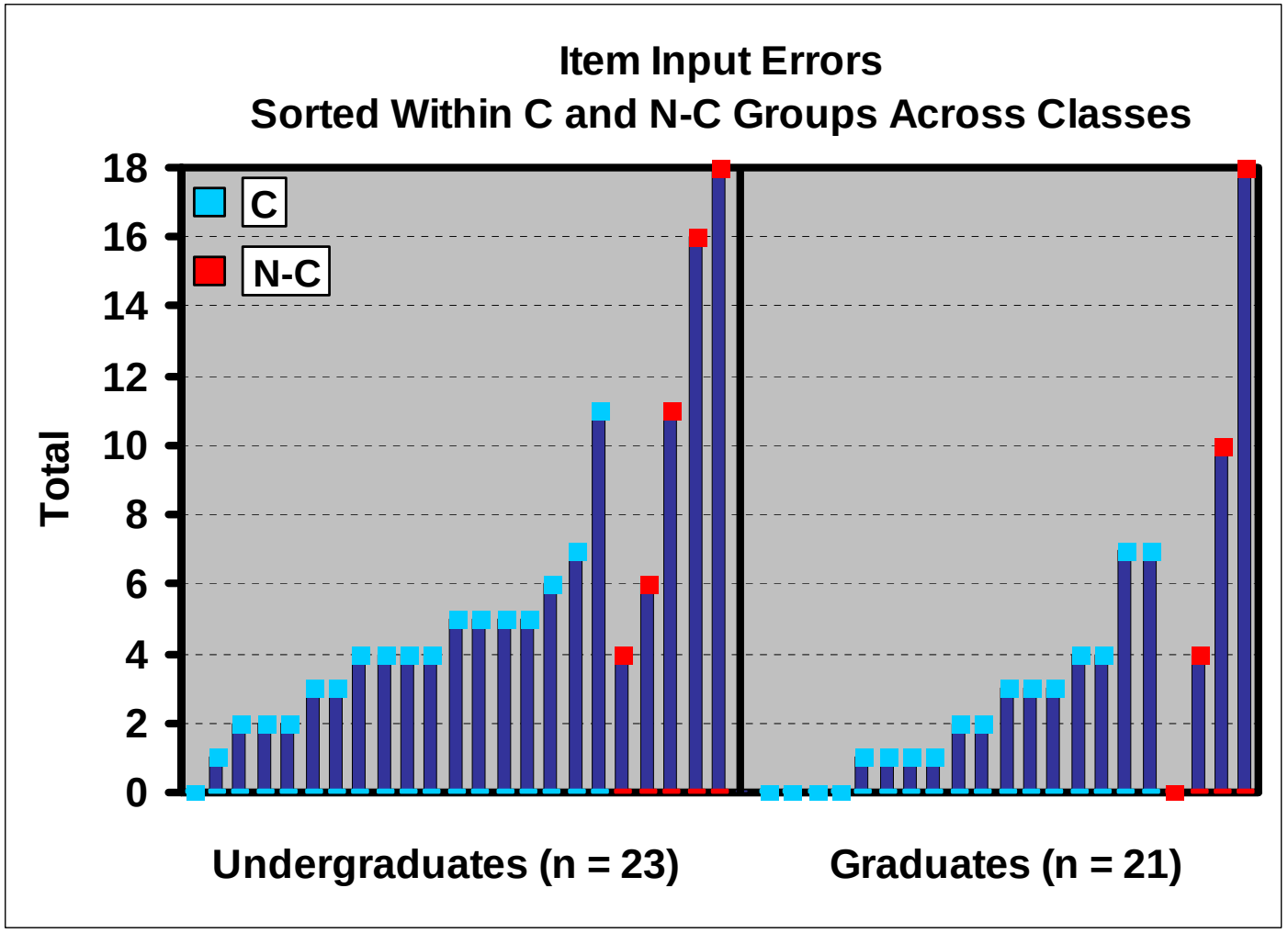
U vs. G: Kruskal-Wallis Chi-Square = 0.15, p > .05.

C vs. N-C: Kruskal-Wallis Chi-Square = 1.65, p > .05.



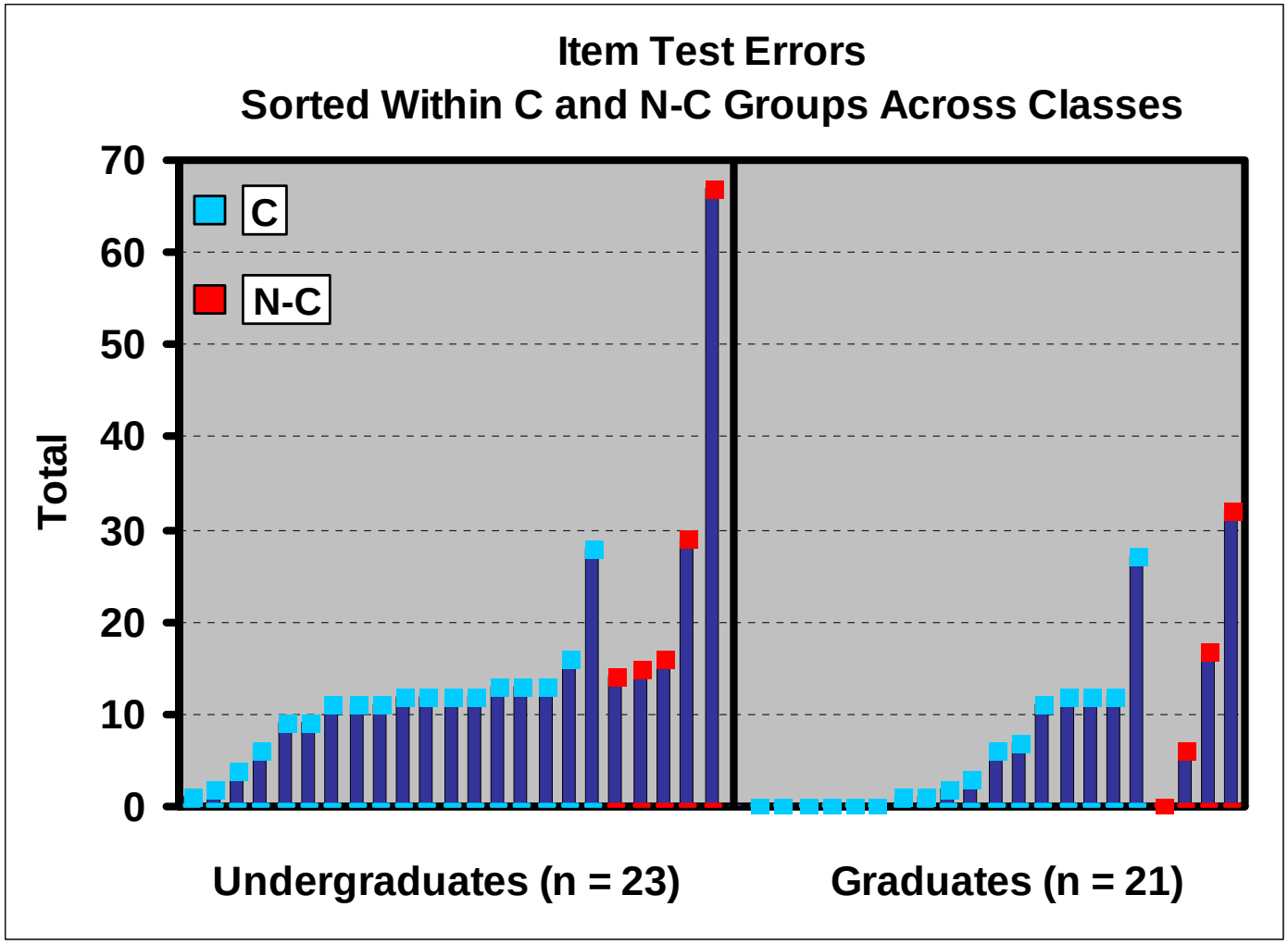
U vs. G: Kruskal-Wallis Chi-Square = 3.99, p < .05.

C vs. N-C: Kruskal-Wallis Chi-Square = 5.07, p < .05.



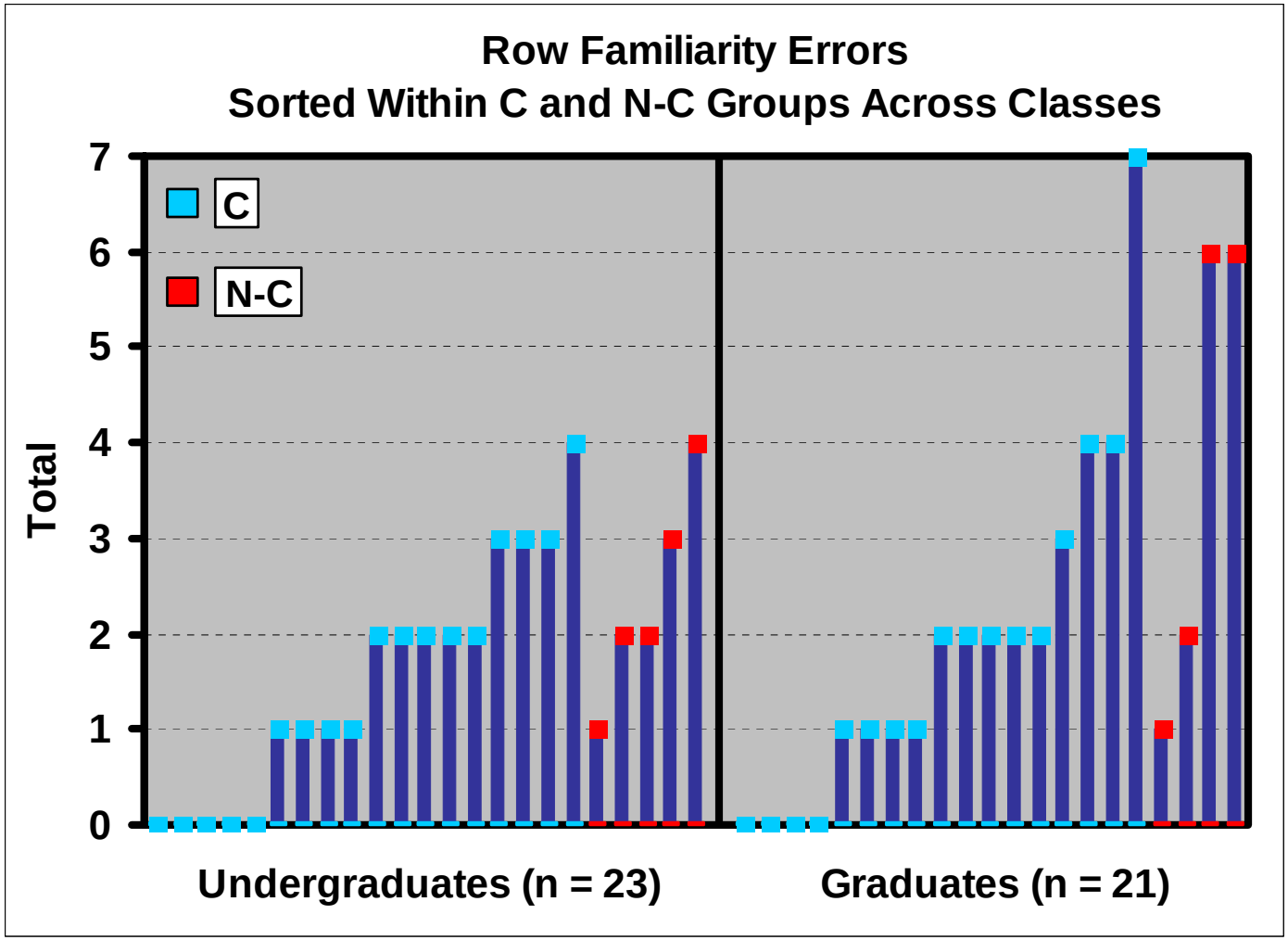
U vs. G: Kruskal-Wallis Chi-Square = 5.71, p < .05.

C vs. N-C: Kruskal-Wallis Chi-Square = 7.75, p < .01.



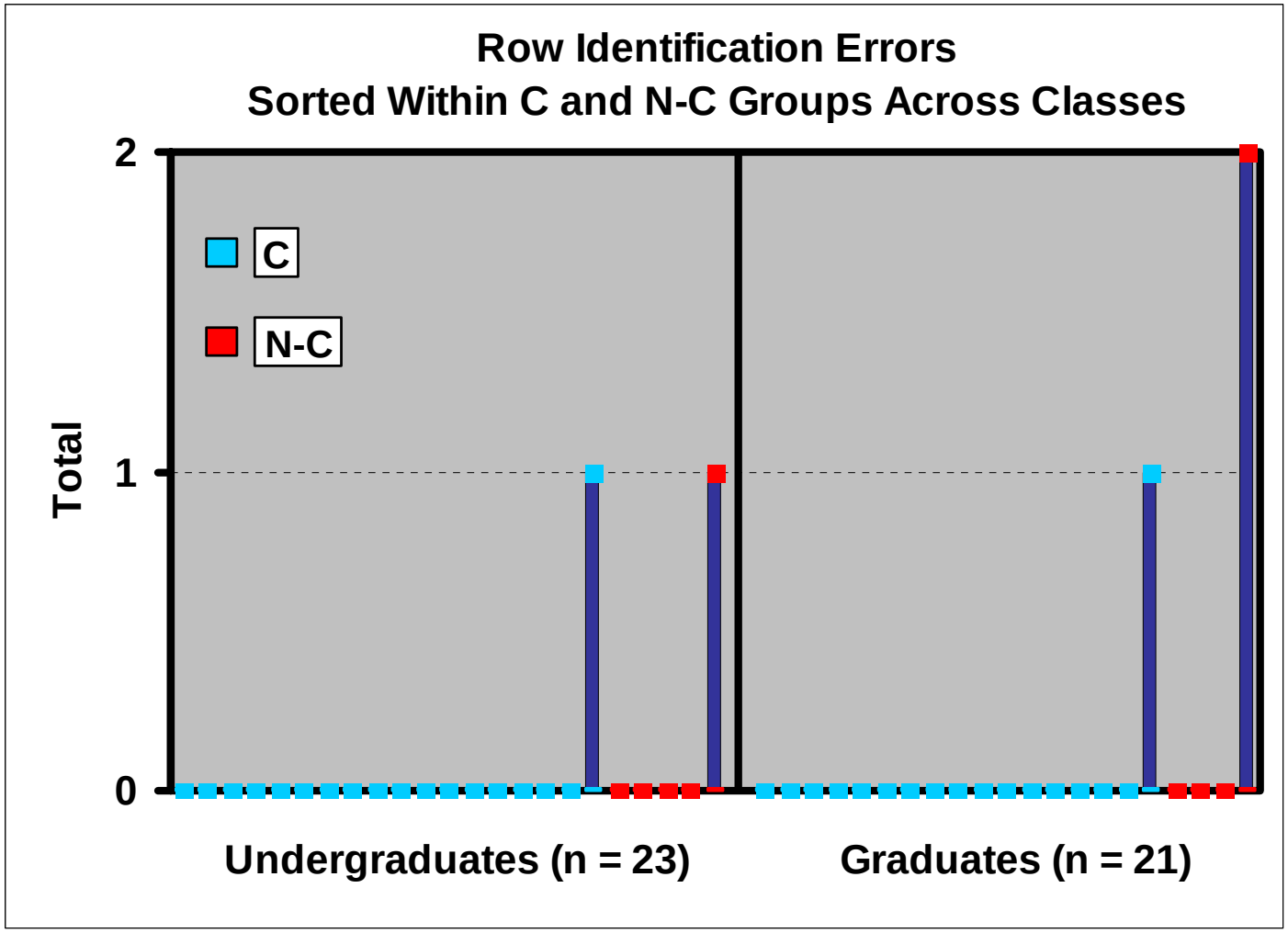
U vs. G: Kruskal-Wallis Chi-Square = 8.50, p < .01.

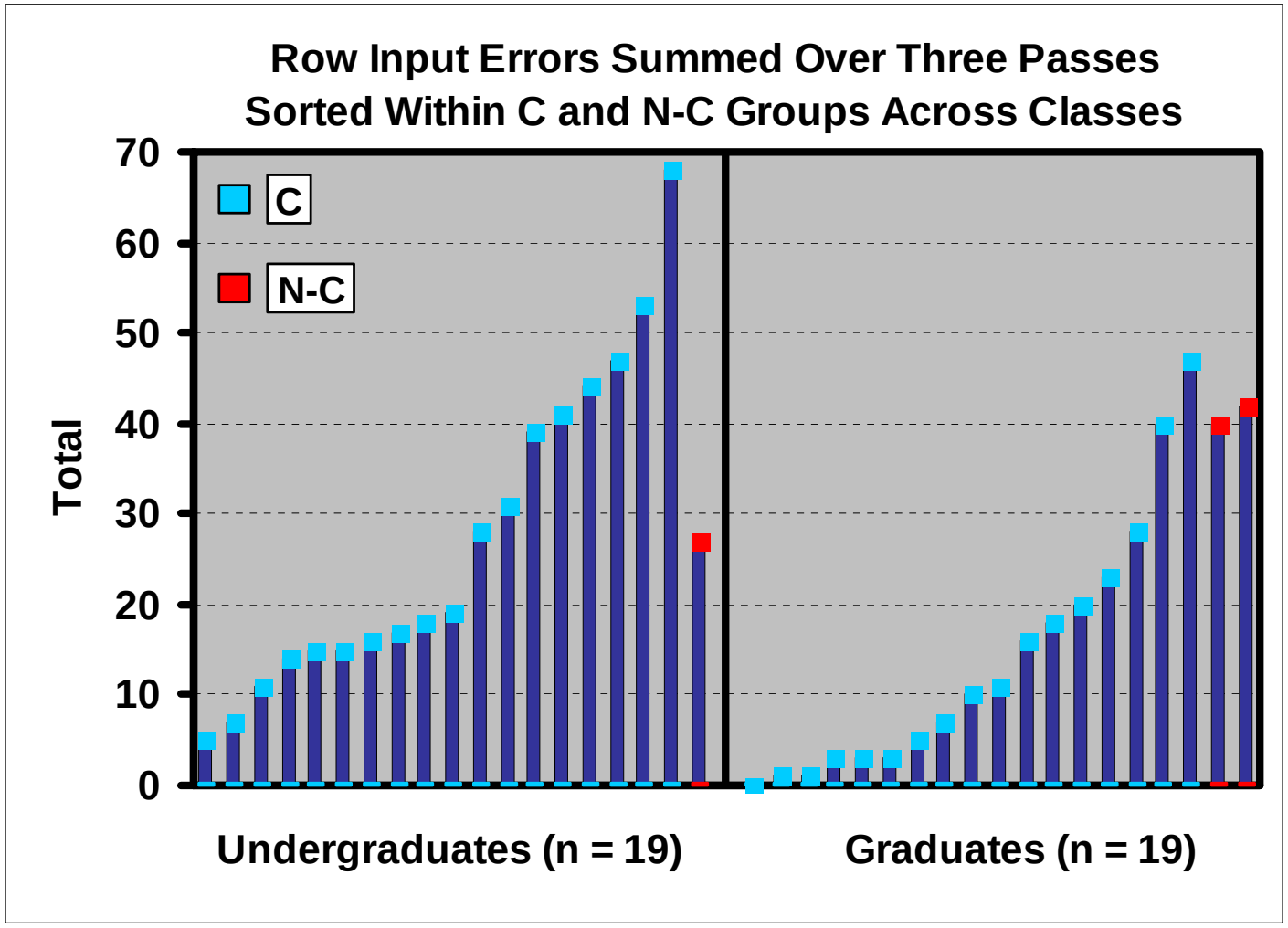
C vs. N-C: Kruskal-Wallis Chi-Square = 7.55, p < .01.



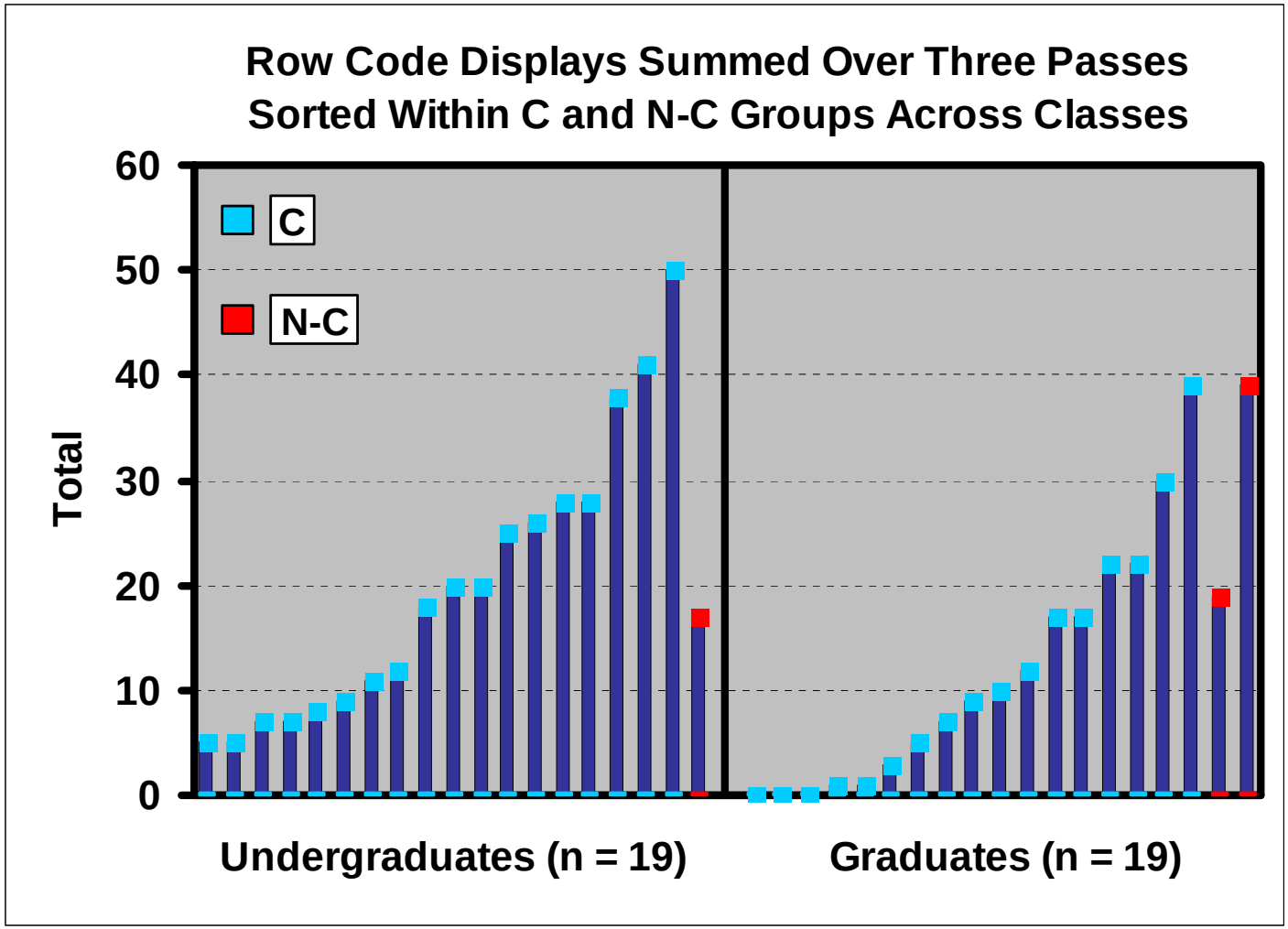
U vs. G: Kruskal-Wallis Chi-Square = 0.21, p > .05.

C vs. N-C: Kruskal-Wallis Chi-Square = 3.64, p > .05.


Row Identification Errors
Sorted Within C and N-C Groups Across Classes
Total
2
1
0
C
N-C
Undergraduates (n = 23)
Graduates (n = 21)

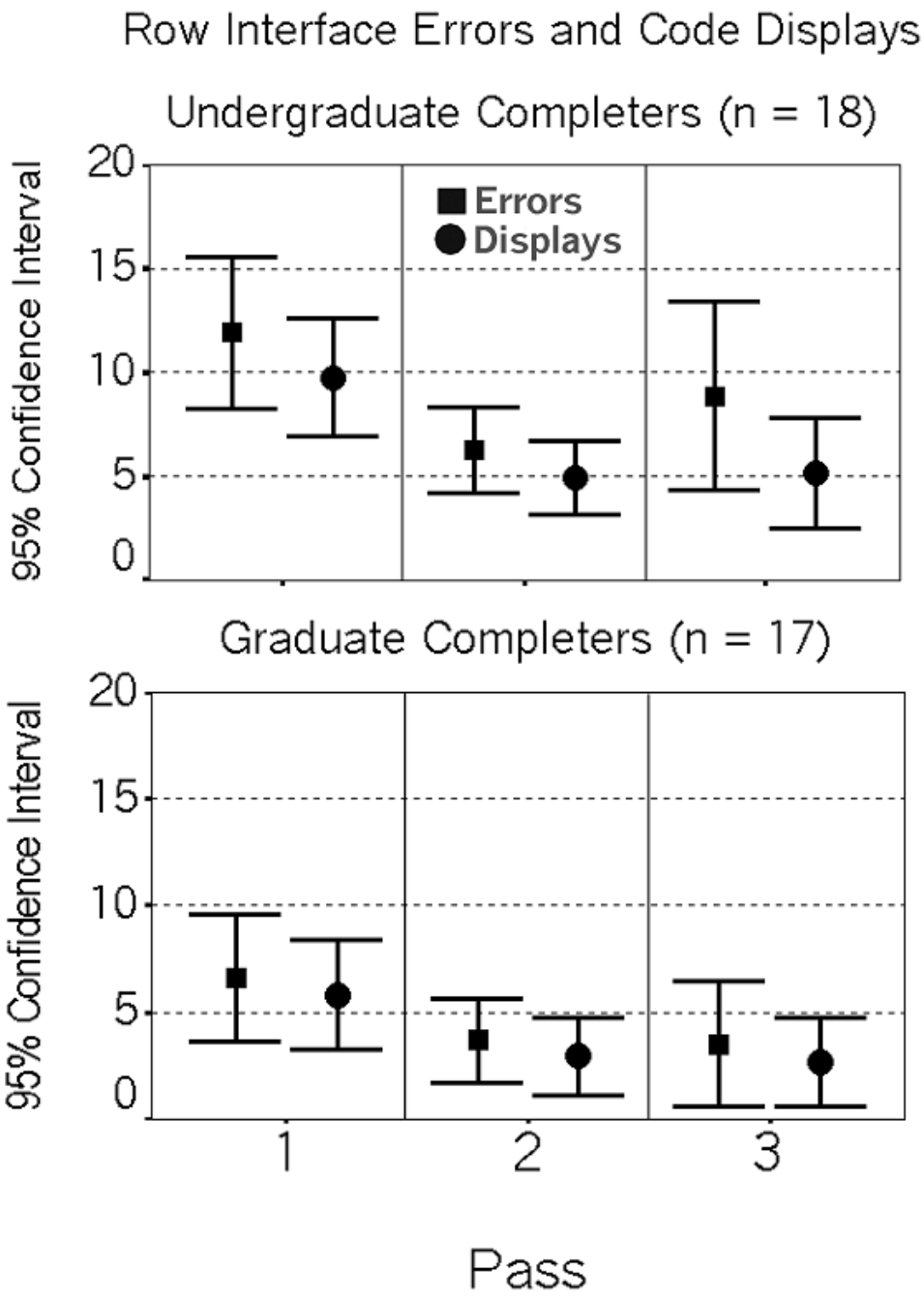


U vs. G: Kruskal-Wallis Chi Square = 3.95, p < .05.

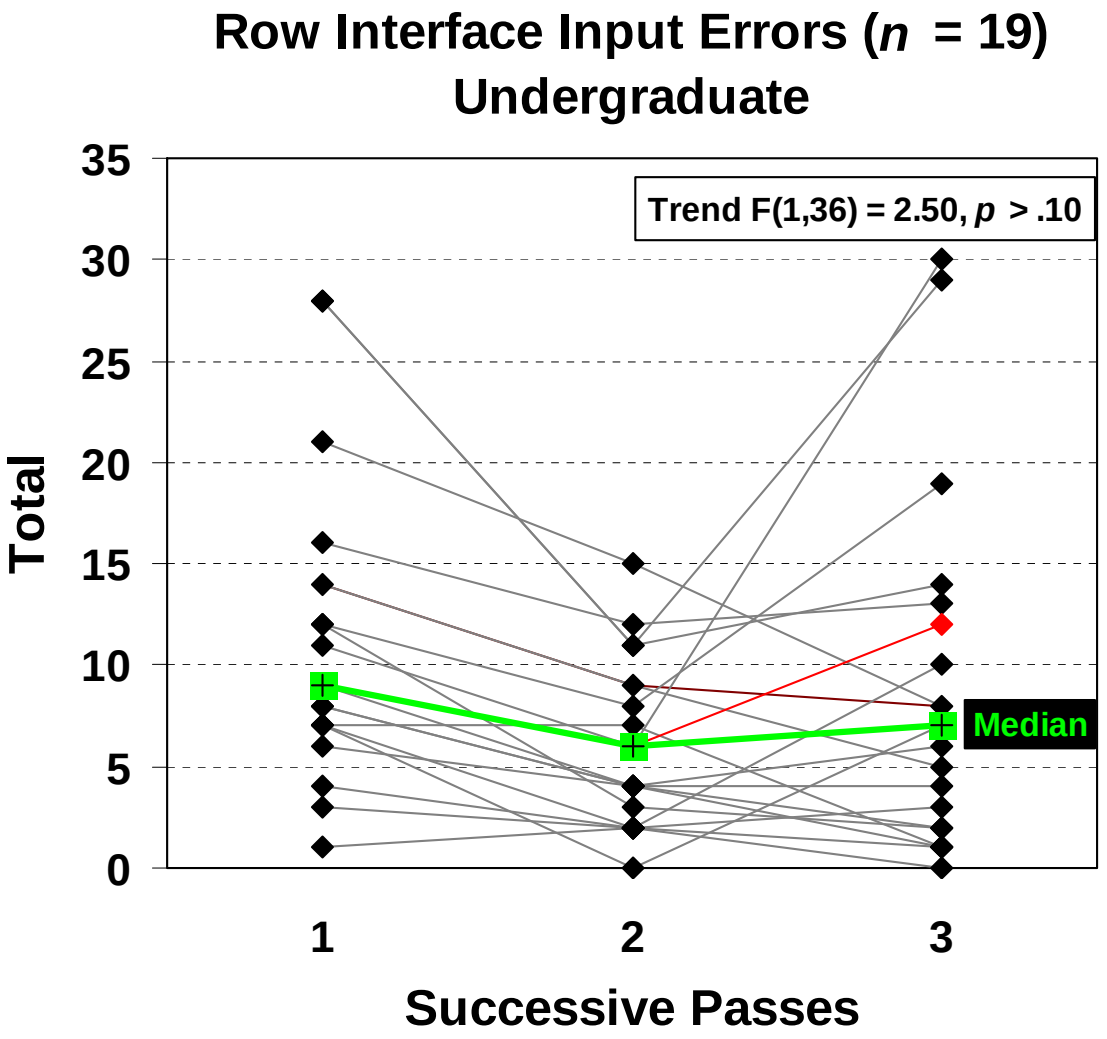


U vs. G: Kruskal-Wallis Chi-Square = 2.73, p > .05.

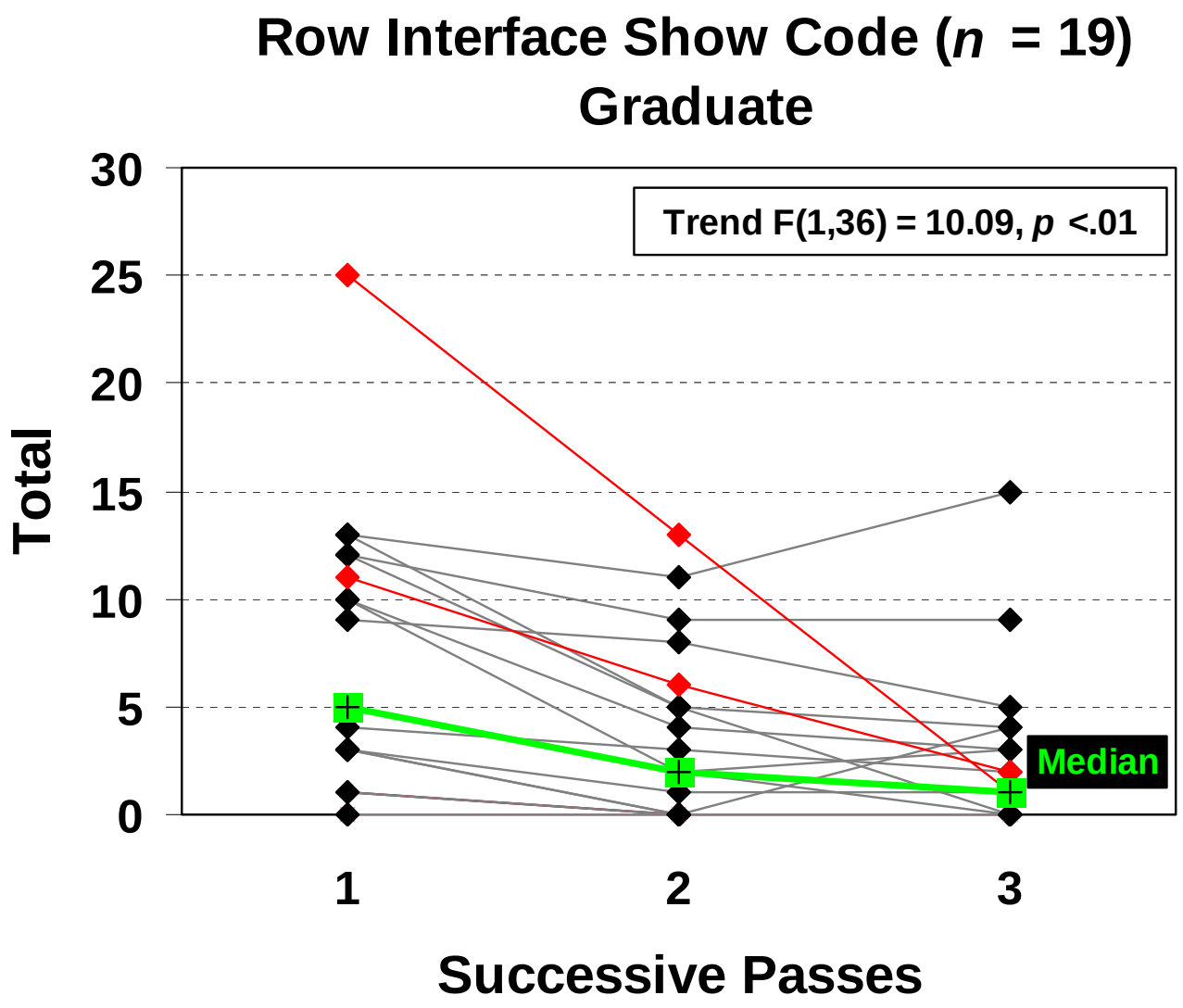
- Main effects of Class: Errors and Displays
- Linear trend: All except U errors.

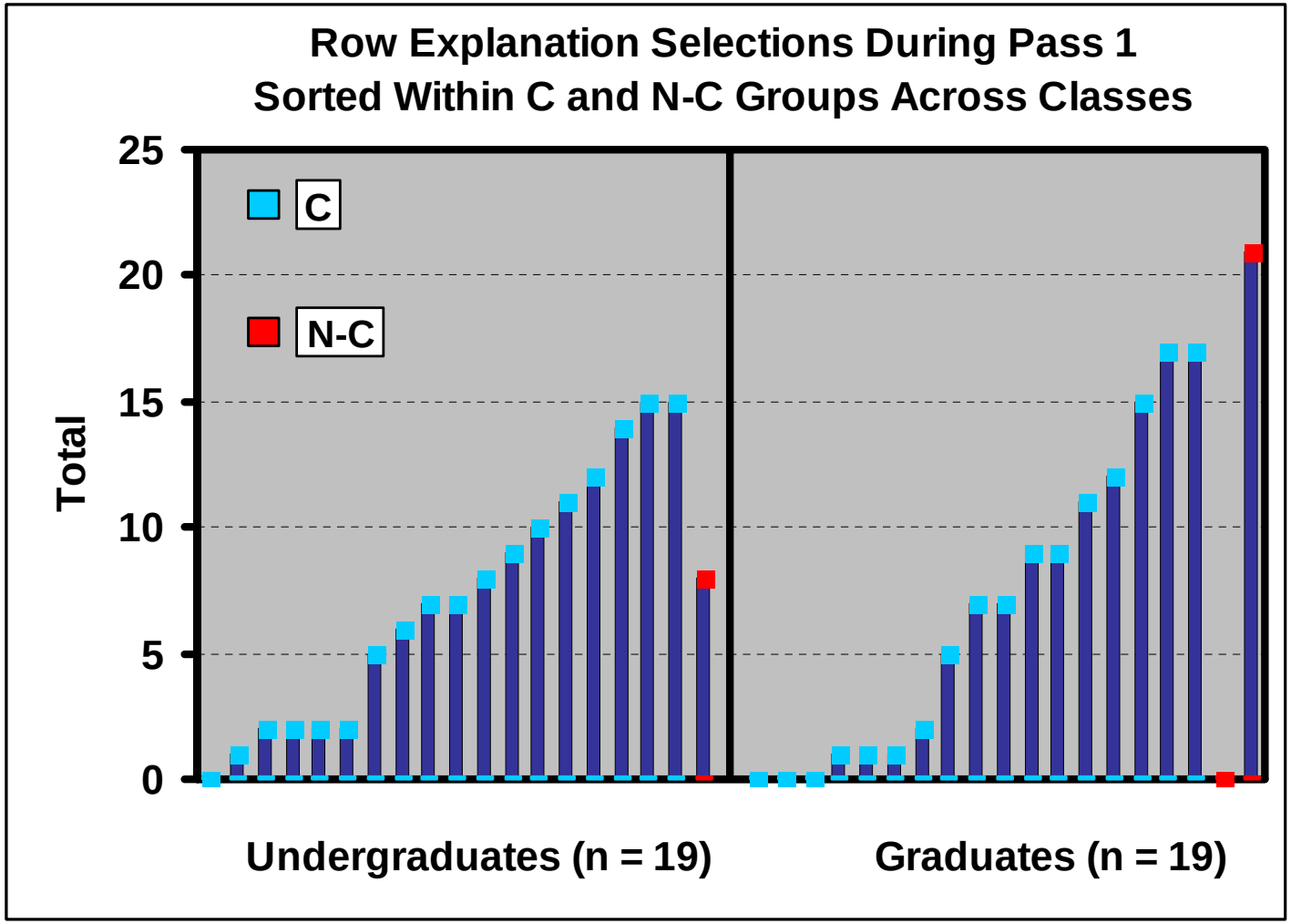


Row Input Errors: Undergraduate

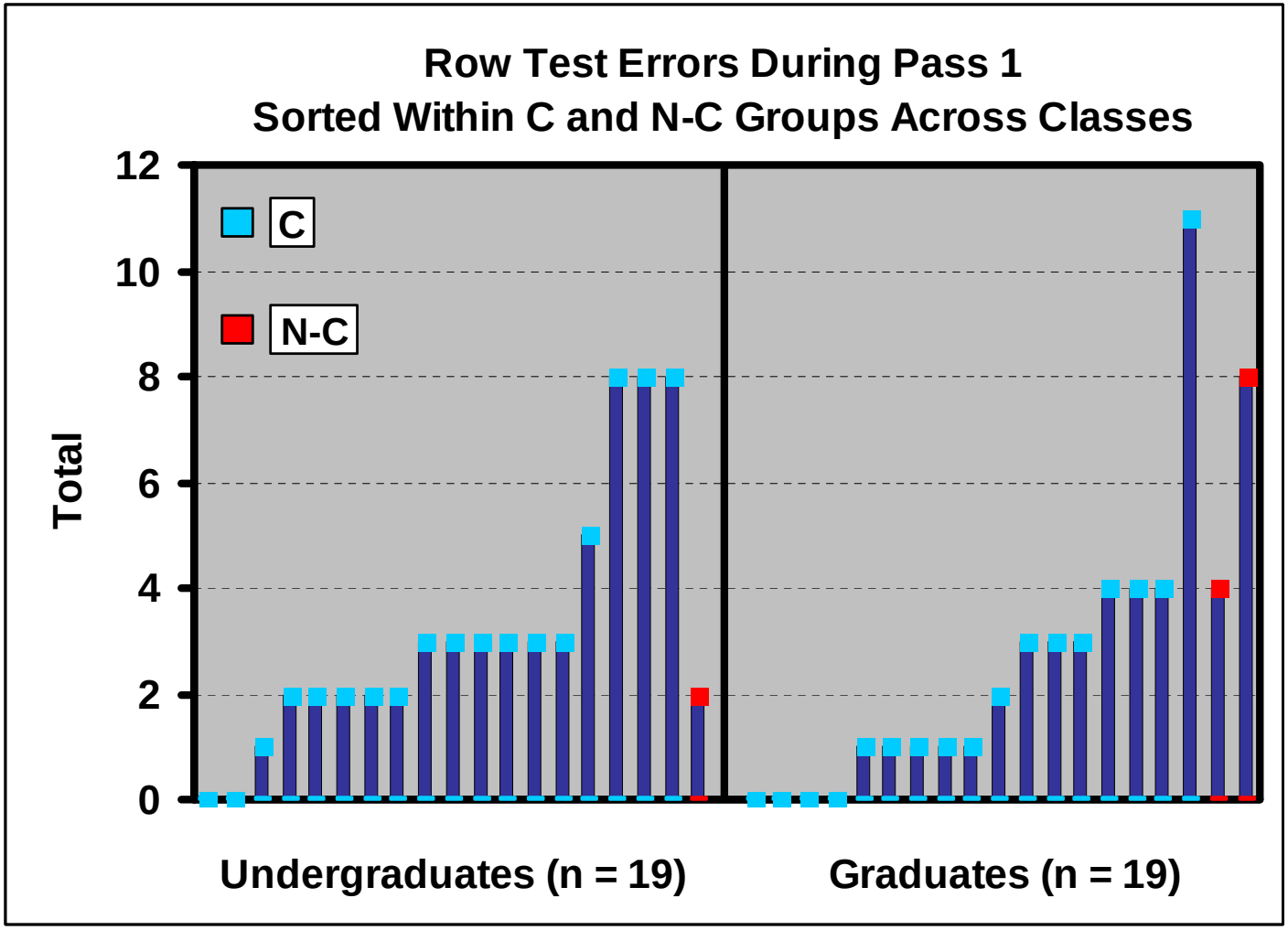


Show Code on a Row: Graduate

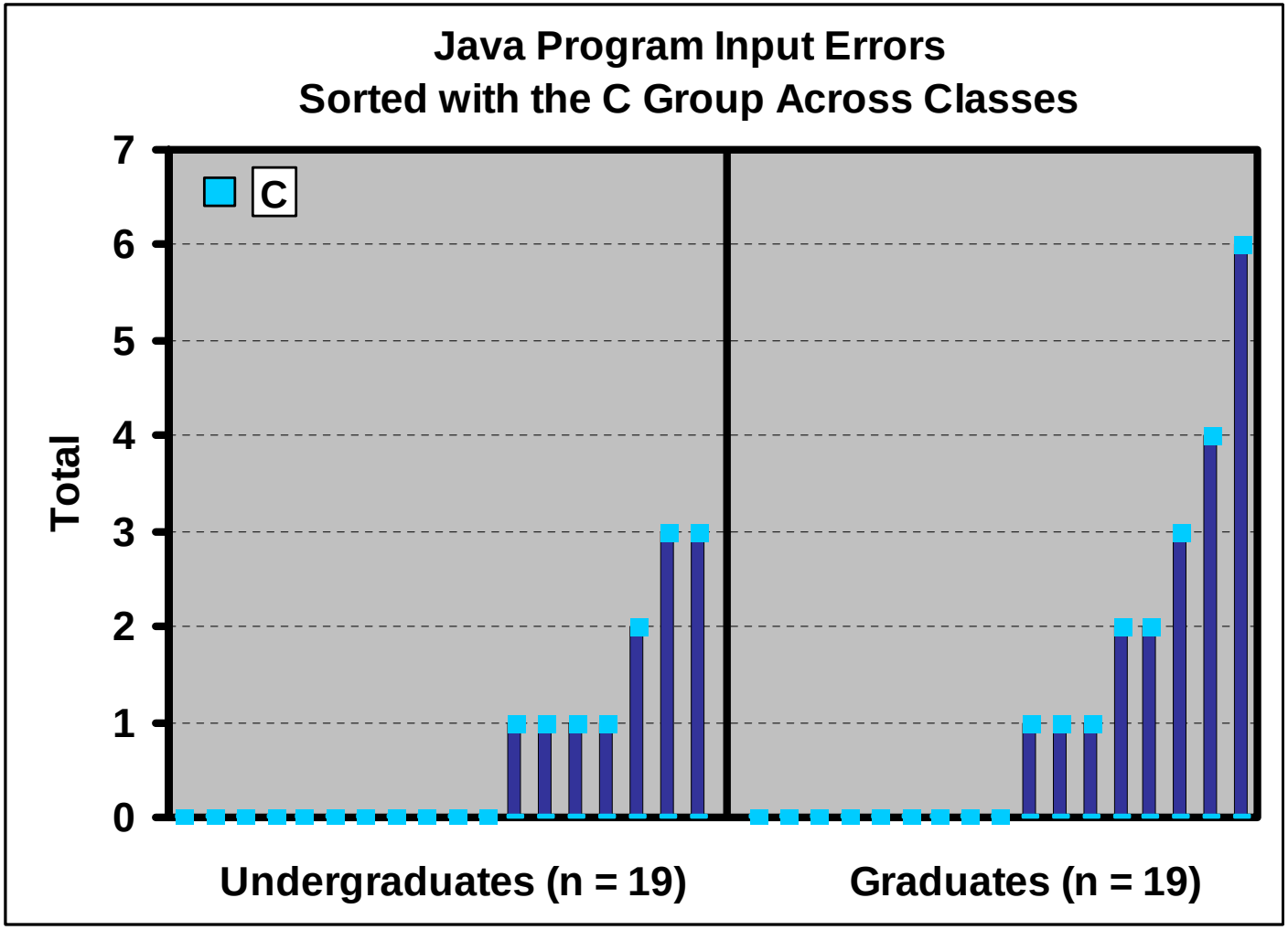




U vs. G: Kruskal-Wallis Chi-Square = 0.25, p > .05.



U vs. G: Kruskal-Wallis Chi-Square = 0.57, p > .05.



U vs. G: Kruskal-Wallis Chi-Square = 0.52, p > .05.

Summer and Fall 2002

- Pre-Training/Post-Training Assessment
- Graduate students
- Rule-Based questions
- Row questions

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Rule-Based Question #1

- Which of the following lines most likely would be used to reference Frame.class, which is a class file built-in to Java?

1. import java.awt.frame;
2. import java.awt.Frame.class;
3. import java.awt.Frame;
4. import java.awt.frame.class;
5. Not ready to answer.

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Rule-Based Question #2

- Which of the following lines most likely would be used to construct an instance of a Button class

1. myButton = new Button.class(“Hello”);
2. myButton = new Button(“Hello”);
3. myButton = button.class(“Hello”);
4. myButton = Button(“Hello”);
5. Not ready to answer.

Rule-Based Question #3

- Which of the following lines most likely would be used to add a Button object to a container?

1. Add(an instance name);
2. Add(a class name);
3. add(a class name);
4. add(an instance name);
5. Not ready to answer.

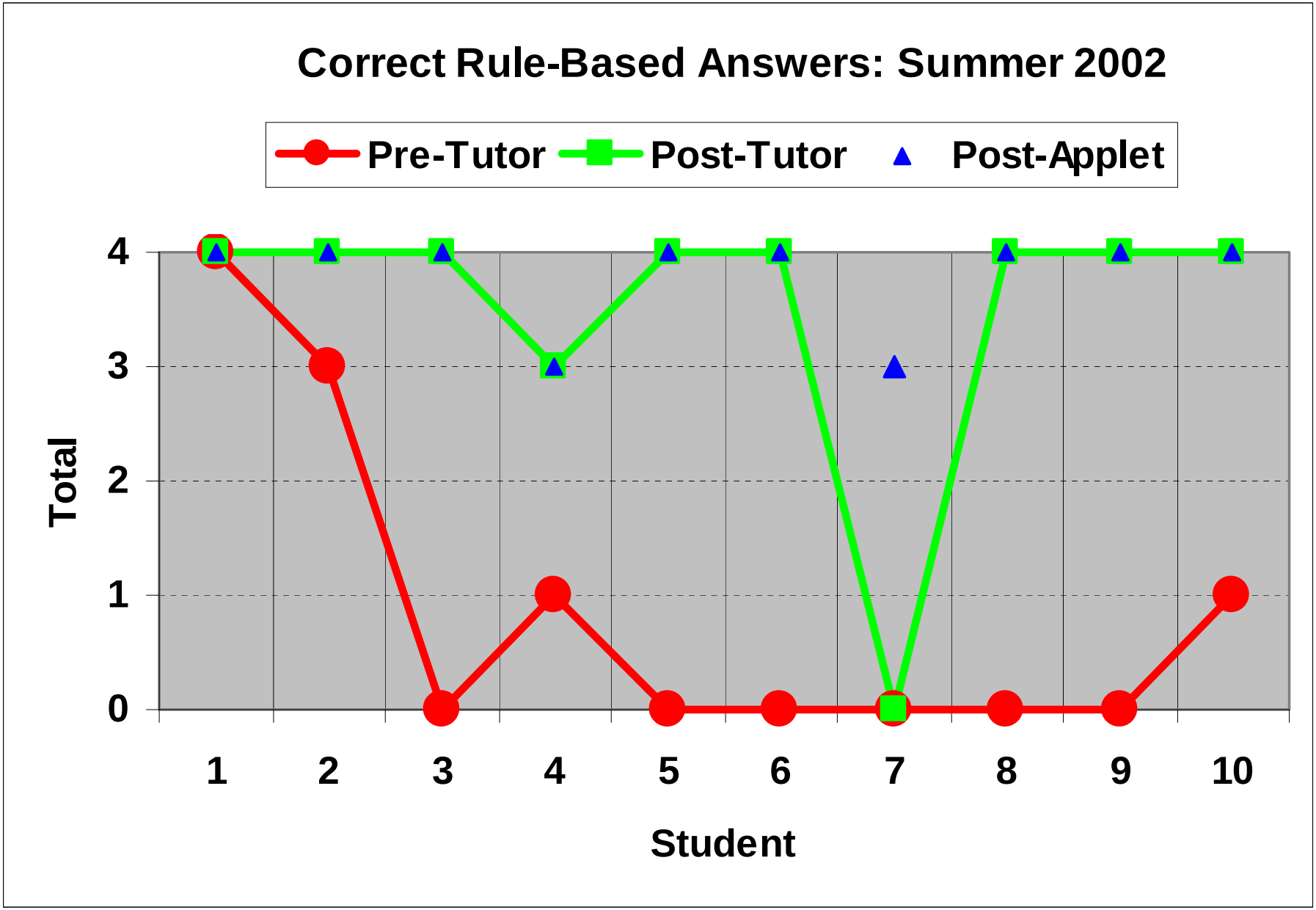
UMBC
AN HONORS UNIVERSITY IN MARYLAND

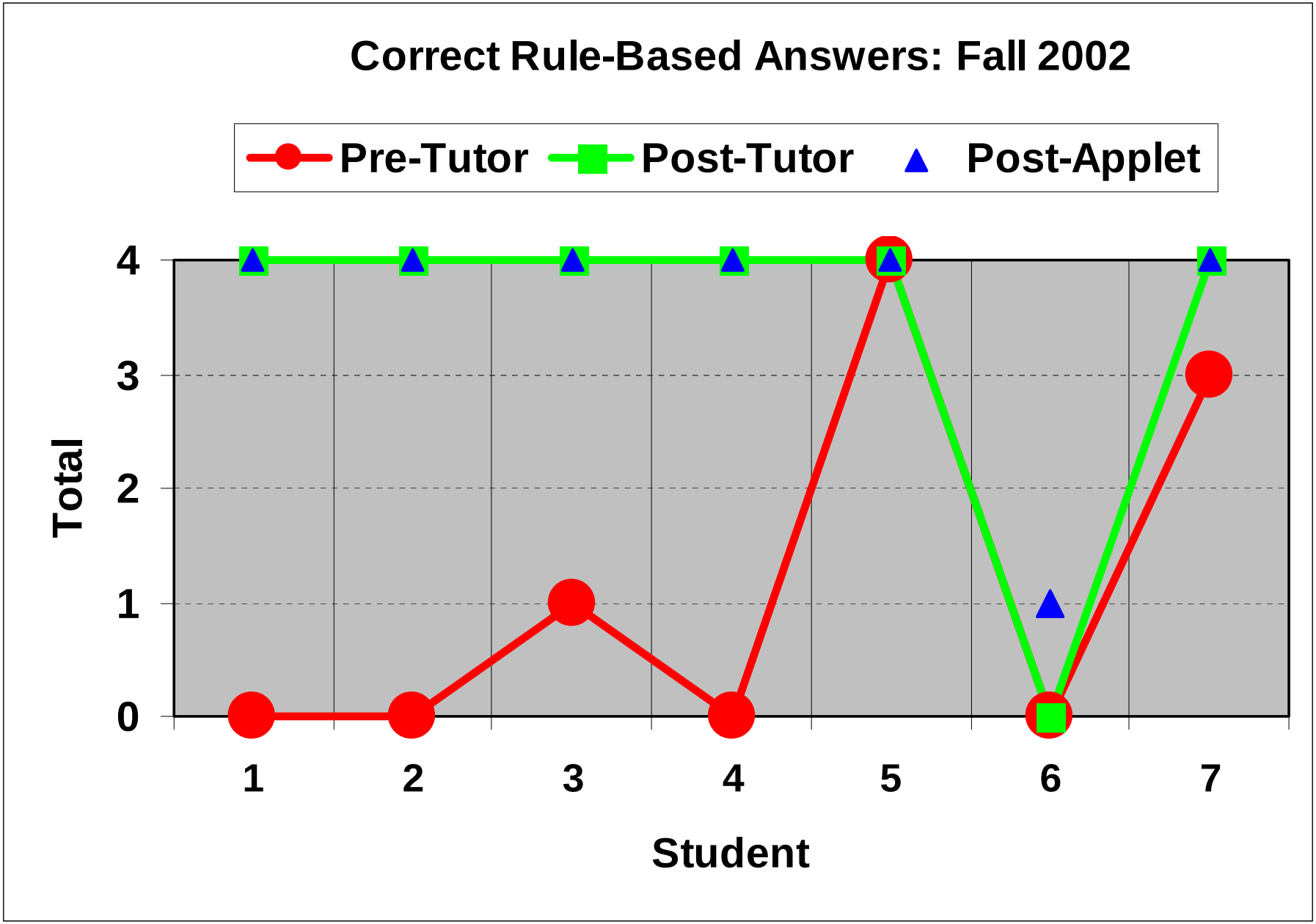
Rule-Based Question #4

- Which of the following lines most likely overrides a method that is contained in the Applet.class file?

1. public void stop(){ lines of Java code here }
2. public void Stop{} { lines of Java code here }
3. Public void Stop() (lines of Java code here)
4. Public void stop() { lines of Java code here }
5. Not ready to answer.

UMBC
AN HONORS UNIVERSITY IN MARYLAND


Correct Rule-Based Answers: Summer 2002
Pre-Tutor
Post-Tutor
Post-Applet
Total
0
1
2
3
4
1
2
3
4
5
6
7
8
9
10
Student


Correct Rule-Based Answers: Fall 2002
Pre-Tutor
Post-Tutor
Post-Applet
Total
0
1
2
3
4
1
2
3
4
5
6
7
Student

The below is the Java program that you will learn or have learned, and it is organized into ten rows of code. Answer the ten questions below as best you can at this point in your learning. Please circle your choice of answer for each of the ten multiple-choice questions.

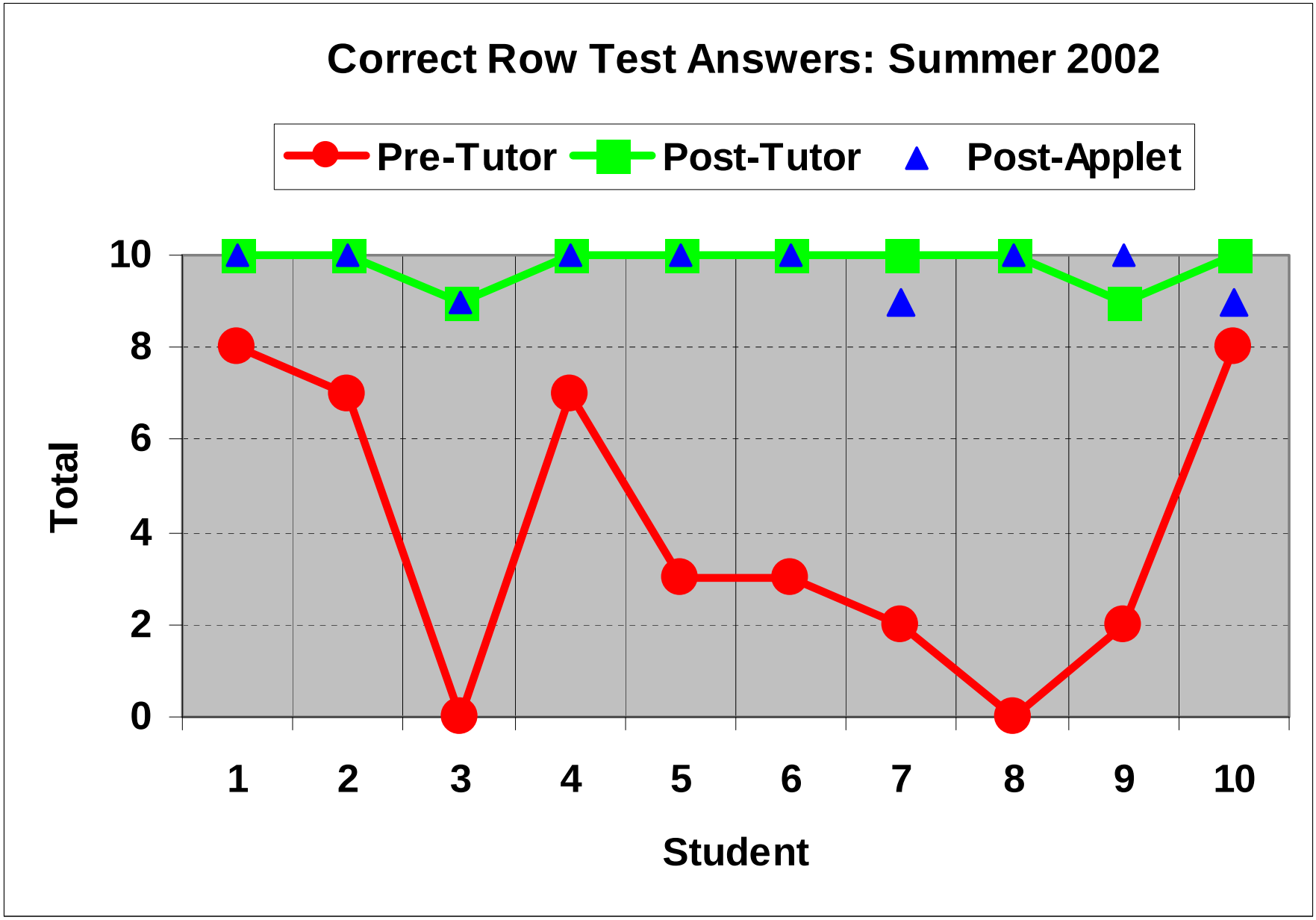
```
Row 1:   import java.applet.Applet;
Row 2:   import java.awt.Label;
Row 3:   public class MyProgram extends Applet {
Row 4:   Label myLabel;
Row 5:   public void init() {
Row 6:   myLabel = new Label(\"This is my first program.\");
Row 7:   add(myLabel);
Row 8:   myLabel.setVisible(true);
Row 9: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

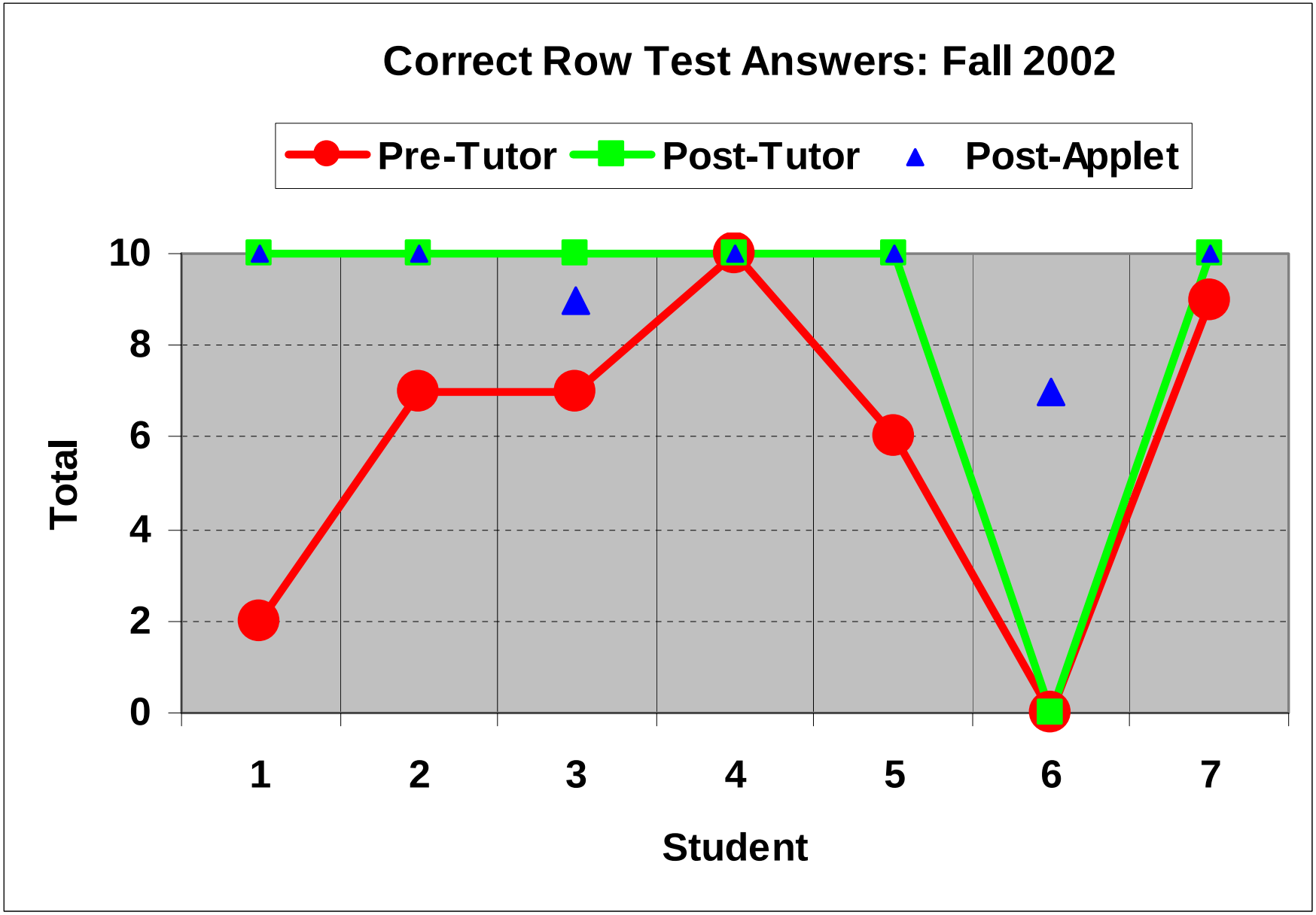
2. What is the overall objective of the code in Row 2?

a. Create a shorthand notation to reference the built-in Label class.
b. Create a shorthand notation to reference the built-in label class.
c. Copy the Abstract Windowing Toolkit.Label directory.
d. The objective is to import the awt.label file.
e. Not ready to answer.

3. What is the overall objective of the code in Row 3?

a. Name a class, MyProgram, that will be a superclass of the Applet class.
b. Name a class, myProgram, that will be a subclass of the Applet class.
c. Override the extends Applet modifiers.
d. Name a class, MyProgram, that will be a subclass of the Applet class.
e. Not ready to answer.


Correct Row Test Answers: Summer 2002
Pre-Tutor
Post-Tutor
Post-Applet
Total
10
8
6
4
2
0
1
2
3
4
5
6
7
8
9
10
Student


Correct Row Test Answers: Fall 2002
Pre-Tutor
Post-Tutor
Post-Applet
Total
10
8
6
4
2
0
1
2
3
4
5
6
7
Student

Observations

- Class differences were evident in the acquisition process.
- Our assessments did not capture the antecedent conditions.
- Learning outcomes appeared equivalent.
  - At least measured on a single occasion
  - Were both classes at the same "steady state"?
- Non-completers could be identified early
  - Adaptive systems
- We need to work on this.

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Conclusions...

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

Conclusions

- ***Software self-efficacy*** can be enhanced by using the tutor.
- ***General rules*** can be acquired by using the tutor.
- ***Students like the tutor*.**
  - Notable exceptions

UMBC
AN HONORS UNIVERSITY IN MARYLAND

How Programmed Instruction Helps...

- Generates a ***history of study behavior*** in students who may lack the study skills and discipline to master the Java code on their own initiative.
- This ***frees the student*** to acquire more advanced levels of skill independent of the support provided by the tutoring system.

UMBC
AN HONORS UNIVERSITY IN MARYLAND

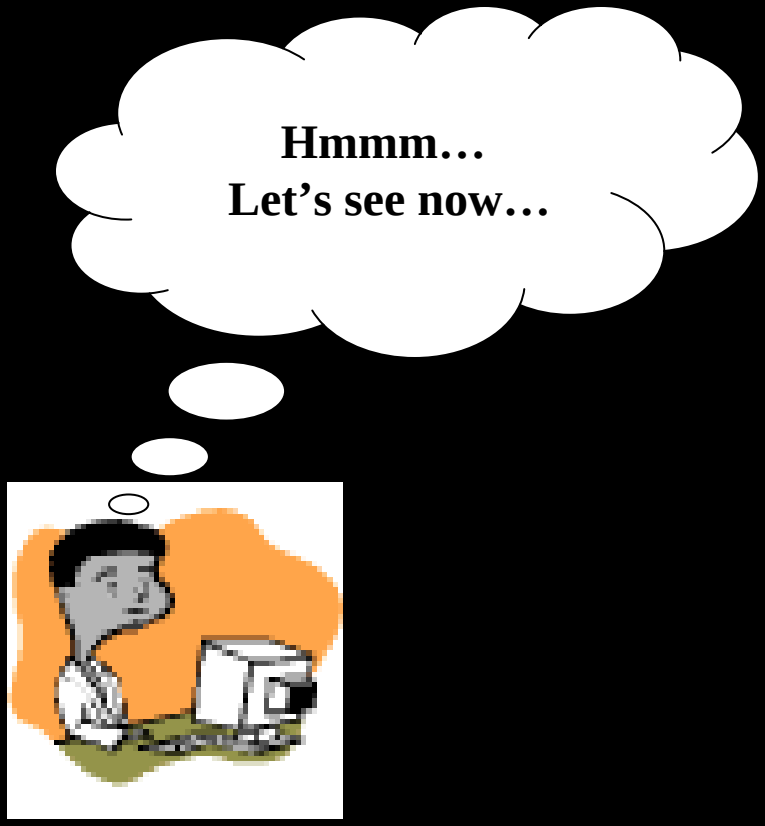
How Programmed Instruction Helps

- ***Software self-efficacy*** is a by-product of effective mastery: enactive mastery.
- It makes information technology ***accessible*** to learners who might otherwise draw away from it.

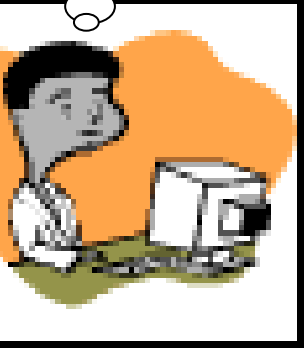
UMBC
AN HONORS UNIVERSITY IN MARYLAND

Bandura: Current symbolic representations can weakly represent future contingencies.

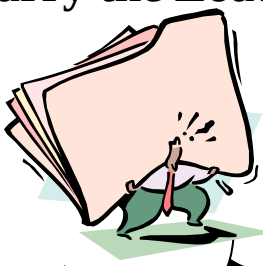
Bandura: Current symbolic representations can weakly represent future contingencies.



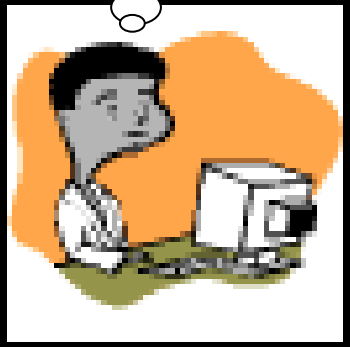
Study



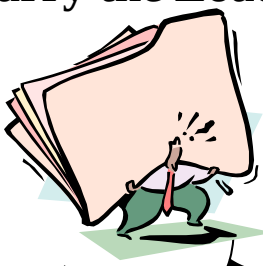
Carry the Load



Study

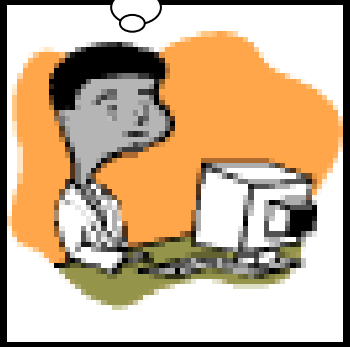


Carry the Load

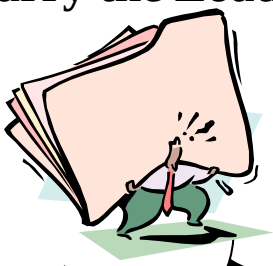


Pull Yourself Along

Study

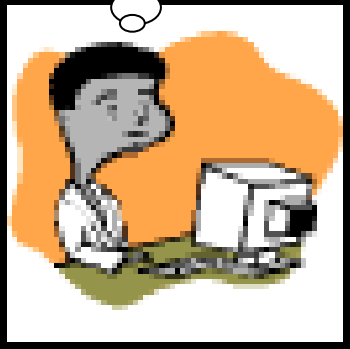


Carry the Load



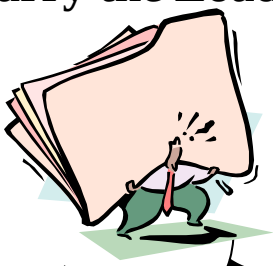
Pull Yourself Along

Study



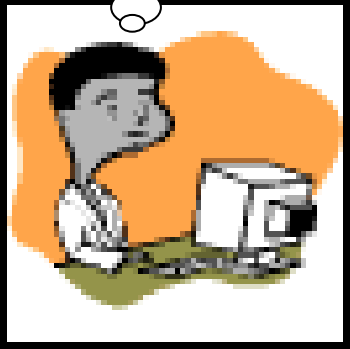
Graduate

Carry the Load



Pull Yourself Along

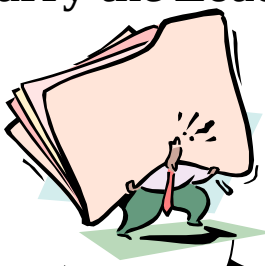
Study



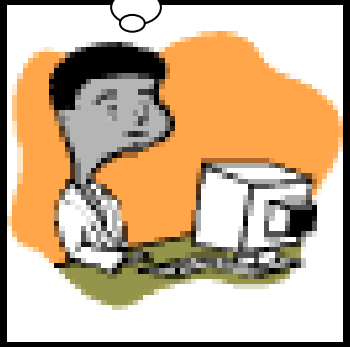
Graduate

Get a Job

Carry the Load



Study



Pull Yourself Along

Graduate

Get the Money!

Get a Job

Tutor URL

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