Combining Programmed Instruction and Collaborative Peer Tutoring to Teach Java™

Henry H. Emurian, Heather K. Holden, & Rachel A. Abarbanel
Information Systems Department
UMBC
1000 Hilltop Circle
Baltimore, Maryland 21250
Forward and Forewarned

• 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.

• 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.
Consequence

This is my first program.
Objective

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

• 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?
  – Programmed instruction + collaborative peer learning
Model

Power Function

Errors

Practice Trials

True Gain
Setting the Stage

Programmed Instruction

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

Interteaching

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.
Introduction

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 yet.

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

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 a JApplet, is started by using MyProgram.html as the target file in the browser URL.

```
<html>
<title>MyProgram.html</title>
<body bgcolor=black>
<center>
<applet code="MyProgram.class" height=300 width=300>
</applet>
</center>
</body>
</html>
```
Java Tutor: Item Learning

practice, time, and exposure to new information. You will be more knowledgeable and skilled after even a single repetition of the tutor.

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

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

public class MyProgram extends JApplet {
    JLabel myLabel;
    public void init() {
        myLabel = new JLabel("This is my first program.");
        myLabel.setVisible(true);
        getContentPane().add(myLabel);
    }
}
```

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. If you can't recall the Java Item when asked to type it, simply press the Enter key with the cursor in the line before the cursor. One technique that works is to type the code and then press the Enter key again. This will give you a new line to type on.

The second line of code is this:

```java
import javax.swing.JLabel;
```

The Java term being taught is highlighted in blue.

```java
import javax.swing.JLabel;
```

The `import` term, when used in the above line, allows you to refer to the built-in Java class, JLabelclass, with a shorthand notation. The reason to use the `import` keyword is because the JLabel class file is not located in your current directory and it is needed to write your program.

For example, to use the built-in Java class file, JLabelclass, in your program, you write:

```java
import javax.swing.JLabel;
```

at the beginning of the program. Then you may use JLabel by itself in the program, as explained later in the tutor, rather than having to use javax.swing.JLabel in the program. The compiler will then be able to find the JLabel class file on the system. The `import` keyword, then:

```
which one of the following statements is correct?

- import javax.swing.JApplet;
- import java.swing.JApplet;
- import java.swing.JButton;
```

- import java.swing.JButton;
Item Test Outcomes

What is the purpose of `javax.swing.JApplet` at this point in the program?

- This is a reference to the `JApplet` class file, `JApplet.class`, located in the `swing` package on the system.
- Copy `JApplet` class into the current directory.
- This is a reference to the `JApplet` class file, `JApplet.class`, located in the `swing` package on the system.

Test Outcome

Your choice was correct.

The `import` keyword is used with `javax.swing.JApplet` to allow the programmer to use `JApplet` by itself later in the program, and the compiler will be able to find the `JApplet.class` file, which is located in a different directory (or package) on the system.

Notice that both `javax` and `swing` are all lowercase letters, and `JApplet`, which is a class, begins with a capital letter. The name of a class always begins with a capital letter. That is an important rule to remember.
Multiple Exemplars

import javax.swing.JTextField;

Your choice was correct: JTextField;

The answer is the only answer that begins with a capital letter and ends with a ; mark. You know that JTextField is a class because it begins with a capital letter. A complete Java statement must end with a ; mark. The import keyword is followed by a path that ends with a class name. The ; mark completes the statement in this line of code. Given the rule about the name of a class, other classes in the swing package might also be used there.
Examples are below:
JButton;
JFrame;
 JPanel;
 JLabel;
 JScrollPane;
Incorrect:
View
Correct
Input

Correct
import javax.swing.JApplet;
import javax.swing.JLabel;
public class MyProgram extends JApplet {
    JLabel myLabel;
    public void init() {
        myLabel = new JLabel("This is my first program.");
        myLabel.setVisible(true);
        getContentPane().add(myLabel);
    }
}
import javax.swing.JApplet;

***** xxxxxxxxxxxxxxxxxxxxxxxxx
Program Interface
import javax.swing.JApplet;
import javax.swing.JLabel;
public class MyProgram extends JApplet {
    JLabel myLabel;
    public void init() {
        myLabel = new JLabel("This is my first program.");
        myLabel.setVisible(true);
        getContentPane().add(myLabel);
    }
}
Type the program in the white space below, and select the Submit button using the Tab key. Please do not use notes. Try to enter the program correctly. If you can’t remember the program, just select Submit. Code again.

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

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.

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

• Contrasts with between-group studies concerned with effect size differences – **True Gain**
### No Pain, No *(true)* Gain

<table>
<thead>
<tr>
<th><strong>IS 247J:</strong> Introductory Programming Using Java</th>
<th><strong>Spring 2005</strong> <em>(n = 22)</em></th>
<th><strong>Fall 2005</strong> <em>(n = 12)</em></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>150-min Classes</strong></td>
<td><strong>Questionnaires:</strong> SSE, Rules, Tutor Items, and Tutor Lines</td>
<td><strong>Questionnaires:</strong> SSE, Rules, Java Scale</td>
</tr>
</tbody>
</table>
| **Class 1** 2/2/2005 | 1. Pre-Tutor Questionnaires | 1. Course Orientation  
2. **Tutor:** 120 Minutes  
2. Pre-Tutor Questionnaires |
| **Homework** | Access to Tutor Study Manual | **Homework** |
| **Class 2** 2/9/2005 | 1. **Interteaching:**  
45 min  
2. Post-Learning Questionnaires  
3. Lecture  
4. Run the Program | **Class 2** 9/7/2005 |
| **Class 3** 2/16/2005 | 1. Final Questionnaires  
– Test credit | **Class 3** 9/12/2005 |
| | | 1. **Interteaching:**  
45 min  
2. Post-Interteaching Questionnaires |
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?
JApplet
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?
JLabel
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 Question: Spring 2005 (12 questions, 4 choices)

- Which of the following lines would most likely add a JTextField object to a JPanel object?
  1. JPanel.add(JTextField);
  2. JPanel.add(myJTextField);
  3. myJPanel.add(JTextField);
  4. myJPanel2.add(myJTextField2);

How confident are you that you selected the correct answer?
1 = No confidence ... 10 = Total confidence.
Enter a number: _____
Rules Test Question: Fall 2005 (14 questions, 5 choices)

• Which of the following lines most likely would be used to add a Checkbox object to a content pane?

1. getContentPane().Add(myCheckBox);
2. container.Add(CheckboxObject);
3. add(container.Checkbox);
4. getContentPane().add(myBox);
5. add(myCheckBox);

How confident are you that you selected the correct answer?
1 = No confidence ... 10 = Total confidence.
Enter a number: ____
The below questions are based on the design of the Java programming language and associated conventions of the language. **Functionality** refers to the effects of an item in a Java program. **Classification** refers to: *keyword, identifier, method, separator, and operator.* Give the most informed rating that you can at this point in your understanding of Java. Give one number for Functionality and one number for Classification.

1. How similar to each other are the following two items in terms of functionality and classification?

   (1) `import`   (2) `new`

**Functionality:** Not Similar  1  2  3  4  5  6  7  8  9  10 Highly Similar
Enter a number here:

**Classification:** Not Similar  1  2  3  4  5  6  7  8  9  10 Highly Similar
Enter a number here:
Item Test Example: 34 Items

Which of the following is correct, given the line:

```
myLabel = new JLabel("This is my first program.");
```

1. `myLabel` is being declared as an instance of the `JLabel` class.
2. `JLabel` is being constructed as an instance of the `Label` class.
3. `myLabel` is being constructed as an instance of the `JLabel` class.
4. `myLabel` is set to equal `JLabel`.
5. `myLabel` is the name of a built-in Java class file named `myLabel.class`.
Which of the following is a correct Java statement?

1. import javax.swing.JLabel;
2. import javax.swing.JLabel
3. javax.swing.JLabel
4. javax.swing.JLabel;
5. import javax.swing.JLabel;
Interteaching Instructions

• IS 247
• Interteaching Report #1
• Your name ________________________________________ Date ________
• Your partner’s name: _______________________________
• 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.

1. import javax.swing.JApplet;
2. import javax.swing.JLabel;
3. public class MyProgram extends JApplet {
4. JLabel myLabel;
5. public void init() {
6. myLabel = new JLabel("This is my first program.");
7. myLabel.setVisible(true);
8. getContentPane().add(myLabel);
9. }
10. }
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 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: _____.

If you have questions that need to be answered in the lecture today, please post them on the Discussion Board now.
Interteachers in Action!
Confidence in Rules Test Answer Accuracy

**Spring 2005**

Confidence in Accuracy of Rules Answers

1 = No Confidence ... 10 = Total Confidence

**Fall 2005**

Confidence in Accuracy of Rules Answers

1 = No Confidence ... 10 = Total Confidence

- Pre-Tutor
- Post-Tutor
- Post-Learning
- Graded Quiz

- Interteaching

- Lecture

N = 20, 21, 21, 19, 18, 12, 12, 12, 12, 9

Right, Wrong
Classification and Functionality Ratings

Compare: (1) import    (2) new
1 = Not Similar ... 10 = Highly Similar

Compare: (1) jApplet   (2) JApplet
1 = Not Similar ... 10 = Highly Similar
Conclusion

• 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.

• Interteaching may add value.

• The competency attained sets the occasion for advanced learning with enthusiasm.

• Students like the tutor and the interteaching.
Thank you!

- Java Tutor, Questionnaires, Study Manual, and Tutor Source Code:
  - http://nasa1.ifsm.umbc.edu/learnJava/tutorLinks/TutorLinks.html