

A Tutoring System for Parameter Passing in Programming Languages

Harsh Shah, Amruth N. Kumar

Ramapo College of New Jersey

`amruth@ramapo.edu`

Contents

- **Parameter Passing**
 - **The Mechanisms**
 - **The Complications**
- **The Role of Problem-Solving**
- **Our Tutor**
- **Evaluation of the Tutor**
- **Future Work**

Parameter Passing Mechanisms

Students generally familiar with:

- Pass by Value
- Pass by Reference

Students unfamiliar with:

- Pass by Result
- Pass by Value-Result
- Pass by Name

Parameter Passing Mechanisms

Complications:

- Global variables
- Aliasing
- Array elements
- Expressions as array subscripts

Courses where discussed:

- Computer Science I
- Programming Languages

Contents

- Parameter Passing Mechanisms
- **Problem-Solving**
 - **The Role of Problem-Solving**
 - **Tutors for Problem-Solving**
- Our Tutor
- Evaluation of the Tutor
- Future Work

The Role of Problem-Solving

- Specific experience required to recognize general principles [Locke 52]
- Improves retention [Farnsworth 94]
 - But, time-consuming
- Targets application [Bloom 56]
- Few large projects + Many small problems
 - Language education research [Mann 92]

Tutors for Problem-Solving

- Tutors help improve learning by one Sigma [Anderson 95]
- Tutors helped improve learning by 10% in Physics [Kashy 97]
- Our tutors have helped improve learning [Krishna 2000, Kumar 2001]

Contents

- Parameter Passing Mechanisms
- The Role of Problem-Solving
- **Our Tutor**
 - **The Features**
 - **Problem Generation**
 - **Feedback**
 - **Graphical User Interface**
- Evaluation of the Tutor
- Future Work

Features of our Tutor

- Dynamically generates problems
 - Unlimited number of problems
- Automatically solves the problems
 - Grades the learner's answers
- Provides detailed feedback
 - Instant feedback
- Maintain statistics
- Usage – As a supplement, not substitute

Problem Generation

Parameterized instances of pre-defined templates

```
<T1><G1>=<R#>;
```

```
<T1><A1>=<R#>;
```

```
<P0>(){
```

```
    <T1><L1>=<R#>;
```

```
    <P1>(<G1>,<A1>[<L1>]); }
```

```
<P1>(<T1><F1>,<T1><F2>){
```

```
    <G1>=<G1>+<R#>;
```

```
    <A1>[<G1>]=<R#>;
```

```
    <A1>[<F1>]=<R#>;
```

```
    <F1>=<F1>+<R#>;
```

```
    <A1>[<F1>]=<R#>;
```

```
    <F2>=<R#>; }
```

Feedback in the Tutor

Types of feedback:

- Minimal
- Detailed

Detailed feedback in four stages:

- Before function call
- During function call
- During function execution
- During function return

User Interface

The screenshot displays a software interface for a programming problem. It is divided into several sections:

- Code Editor:** Contains C++ code for a function `process` and `main`. A red circle (1) highlights the array `depth` initialization in `main`.
- Variable Initialization:** Shows the state of `depth` (an array of 8 ones) and `principal` (value 3).
- Assumptions:** States that parameters are passed by Value-Result and that actual parameters are evaluated from left to right.
- Quiz Questions:**
 - Question 2: "What is the value of the array `depth` after the execution of `main`?" with a corresponding array input field.
 - Question 3: "What is the value of the variable `principal` after the execution of `main`?" with an input field containing the value 3.
- Feedback:** A section labeled "Create New Problem" shows that the user's answers for `depth` and `principal` were incorrect. It provides the correct answers: `depth` is `{ 1, 1, 3, 2, 1, 3, 2, 1 }` and `principal` is 6. A red circle (4) points to the correct `principal` value.
- Additional Info:** A note states "Parameters are passed by Value Result" and shows the actual parameters when `main` calls `process`: `principal = 3`.
- Buttons:** "Check My Answer" is located at the bottom right.

Contents

- Parameter Passing Mechanisms
- The Role of Problem-Solving
- Our Tutor
- **Evaluation of the Tutor**
 - **The Protocol**
 - **The Results – Cognitive Learning**
 - **The Results – Affective Learning**
- Future Work

Evaluation of the Tutor - Protocol

- As part of graded test in *Programming Languages* course
- Material covered in class before
- All mechanisms in C++ syntax
- Pre-Test → Practice → Post-Test Protocol
 - Pretest and Post-test questions generated by the tutor
 - No access to text or reference material between tests
 - Best of Pre and Post-test scores counted towards course grade

Fall 2001 – Crossover Design

Pass by Reference - Practice Effect

Pass by Value-Result

- Written Pre-Test – 8 min
- Practice with Tutor – 12 min
 - Group A - Minimal Feedback Vs Group B - Detailed Feedback
- Written Post-Test – 8 min;
- Feedback – 5 min

Pass by Name

- Written Pre-Test – 8 min
- Practice with Tutor – 12 min
 - Group A - Detailed Feedback Vs Group B - Minimal Feedback
- Written Post-Test – 8 min;
- Feedback – 5 min

Spring 2002 – Within-Subjects Design

Pass by Reference - Practice Effect

Pass by Result – No Practice

- Written Pre-Test – 8 min
- Written Post-Test – 8 min

Pass by Value-Result – Minimal Feedback

- Written Pre-Test – 8 min
- Practice with Tutor - Minimal Feedback – 12 min
- Written Post-Test – 8 min
- Feedback – 5 min

Pass by Name – Detailed Feedback

- Written Pre-Test – 8 min
- Practice with Tutor - Detailed Feedback – 12 min
- Written Post-Test – 8 min
- Feedback – 5 min

Fall 2001 - Cognitive Learning

Percentage improvement in points per attempted question (N = 8):

	Group A	Group B
Value-Result	4.84%	17.24%
Name	25.96%	10.20%

Spring 2002 - Cognitive Learning

Percentage improvement in points per attempted question (N = 8):

No Practice	Minimal Feedback	Detailed Feedback
7.64% (0.88281)	10.83% (0.21875)	27.60% (0.04688)

Fall 2001 - Affective Learning

Feedback Question	Detailed Feedback	Minimal Feedback
Problems helped illustrate concepts	1.75	2.75
Feedback was clear	2.75	4.00
Feedback was useful	1.75	4.00
Feedback was sufficient	3.00	4.33

Affective Learning contd.

Feedback Question	Detailed Feedback	Minimal Feedback
Helped understand what I knew	2.25	3.33
Helped me learn new material	2.25	3.67
Will use if tutor made available	2.00	3.00
Would like to see such tutors on other topics	1.25	2.50

Spring 2002 - Affective Learning

Feedback Question	Detailed Feedback	Minimal Feedback
Feedback was clear	1.57	2.25
Feedback was useful	1.43	2.88
Feedback was sufficient	1.00	3.13

Contents

- Parameter Passing Mechanisms
- The Role of Problem-Solving
- Our Tutor
- Evaluation of the Tutor
- **Future Work**

Future Work

- Incorporate Error Flagging and Immediate Feedback
- Rewrite feedback to be adaptive
- Set up the tutor to log scores in an online database
- Improve the types of problems randomly generated by the tutor
- Continue to evaluate the tutor's features

Finally...

Tutor *currently* available at

<http://orion.ramapo.edu/~amruth/problets>

Partial support for this work was provided by the National Science Foundation's Course, Curriculum and Laboratory Improvement Program under grant DUE-0088864