

## **2006-2158: PROMOTING ACTIVE LEARNING IN TEACHING THE ORGANIZATION OF PROGRAMMING LANGUAGES COURSE**

### **Jun Luo, Ohio Northern University**

Jun Luo is currently an assistant professor in the Electrical & Computer Engineering and Computer Science Department at Ohio Northern University. He received his B.S., M.S., and Ph.D. in Computer Science from Beijing University of Technology, Tsinghua University, and University of Florida in 1994, 1997, and 2002, respectively. His teaching interests are in data structures and algorithms, databases, compilers, and organization of programming languages.

### **Robert Hovis, Ohio Northern University**

Robert A. Hovis has a Ph.D. in mathematics from Ohio State University and has been a professor of mathematics and computer science for more than 30 years.

# Promoting Active Learning in Teaching the *Organization of Programming Languages Course*

## Abstract

Active learning has been shown to be one of the most efficient and effective teaching methodologies by numerous papers and articles. At Ohio Northern University (ONU), the course, *Organization of Programming Languages* (OPL), is offered to senior students. It is a required course for computer science majors. Two issues arise during the teaching of the course. First, in class, students are reluctant to listen to concepts that they have already learned in prerequisite courses. This happens even though the concepts are now addressed from the angle of the design issues instead of the angle of pure program writing. Second, students are not willing to learn special features that exist in the programming languages that they are not familiar with. To resolve these issues, active learning is introduced into the OPL course. In this paper, we will discuss in detail the implementations of the course and will give an assessment of the implementation.

## 1. Introduction

Active learning is a process by which students must actively be involved in reading, writing, discussion and problem solving. It has been shown to be one of the most efficient and effective teaching methodologies by numerous papers and articles<sup>2, 6, 8, 9</sup>. Active learning has been widely used in classrooms for effective teaching<sup>1, 3, 4, 5, 7, 10</sup>. At the university, OPL is a core, required course in the Computer Science curriculum. It is designed to theoretically investigate programming language constructs and to illustrate the implementations of the constructs in a variety of programming languages. Generally speaking, OPL can be taught either horizontally or vertically. In the horizontal strategy the instructor first discusses the constructs in theory and then demonstrates the concrete implementations in real programming languages. In the vertical strategy the instructor teaches a particular unfamiliar programming language in class and uses the features of the language to explain programming language concepts. Here at ONU, by the time students take OPL, they have already learned several current programming languages from prerequisite courses. Two issues arise during the teaching of the course. First, students are reluctant to listen to concepts that they have already been taught. They do not like repetition, even though the concepts are now addressed from the angle of the design issues instead of the angle of pure program writing. Secondly, students are not motivated to learn special features that exist in the programming languages they do not know. They ask why they must learn a feature of a language that they will likely not use in the future. The students claim that it is difficult for them to understand the features and it is easy for them to forget the meaning of the features because they do not know the language. To resolve these issues, we have incorporated active learning into the OPL lectures. The class is conducted using two threads: 1) the instructor discusses constructs of languages using the horizontal strategy; and 2) all students are required to independently study a language that they are not familiar with and write a paper on that language during the quarter. The students are also required to actively participate in class discussions, using the knowledge they learned through their independent study. The feedback from the

students has been positive. The students consistently report that the way the course is implemented helps them improve their understanding of the course topics. In the remainder of the paper, we will discuss in detail the implementations of the course and will give an assessment of the implementation.

## 2. Implementation of the OPL Course

At ONU, the OPL course is offered every fall quarter. It is taught four times a week for a total of 10 weeks. Each class period lasts 50 minutes. Currently, the course uses the textbook, *Concepts of Programming Languages* by Robert W. Sebesta, 7<sup>th</sup> edition, Addison Wesley Inc., 2005.

The paper and presentation format was introduced into the OPL course several years ago. It has now become an integral component of the course: each student must complete a paper by the end of the quarter, and the paper should meet certain requirements. Each student is also required to give a presentation on the paper in the last week of the quarter. The student's paper and presentation are weighted 20% and 10%, respectively, of his/her final grade for the course. The student who is not able to turn in the paper on time or give the presentation fails the course. The scores for papers and presentations are determined by both the peer evaluations among the students themselves and by the instructor's own judgment.

In addition to the paper and presentation, a student's final grade for the course depends on: a first in-class test, 25%; a second in-class test, 30%; and programming and homework assignments, 15%.

A typical course schedule is as follows:

Week 1: Introduction. In class, the instructor discusses principal programming applications, the history of programming languages, programming language evaluation criteria, programming paradigms, program translation process, and other necessary topics.

At the same time, students choose a programming language for their papers. During the first lecture of the quarter, students are given a list of programming languages. Each language is selected based on four considerations: 1) Students are not familiar with the language, 2) The language has major features of one of the programming paradigms that will be discussed in class, 3) The language plays an important role in the history of programming language evolution, and 4) The language is widely used for certain programming applications, either historically or currently. The list given to the students in the fall quarter of 2005 was as follows: ADA, ALGOL, C#, CLIPS, EIFFEL, FORTRAN, HASKELL, LISP, PASCAL, PERL, PHP, PL/I, SIMULA, and SMALLTALK.

The students are asked to select one language from the list for their independent study during the quarter. The decision has to be made and reported to the instructor at the beginning of the third lecture. To ensure the diversity of paper topics, it is required that no more than two students investigate the same language. The instructor coordinates the topic selection activities.

Along with the list of programming languages, a guideline providing the general requirements of the paper format is also handed out to the students. The requirements include: 1) Each paper will be 8-16 pages, double spaced, with a font size of 12. 2) Each paper must address the following questions: the history and the evolution of the language; the major applications of the language; the programming paradigm of the language; supported data types; control flow mechanisms; the principal advantages and disadvantages of the language compared to other programming languages in the same programming paradigm; and, other major features of the language. 3) Each paper must cite and explain an application coded in the language. The code should reflect the main features of the language. 4) Each paper must cite at least three references, including books, conference or journal papers, or web sites. 5) Each paper should be free of spelling or grammar errors. 6) The paper layout should be effective and visually appealing. All the guidelines will be included in the course syllabus.

Weeks 2 & 3: The procedural programming paradigm. In class, the instructor discusses data types in a variety of languages; binding, scope, visibility, and lifetime of variables; type checking; overloaded operators; flow of control statements; data abstraction and process abstraction; subprogram implementation; and parameter-passing mechanisms.

From time to time, students are asked to provide and explain to the entire class examples from the language they are investigating to illustrate the concepts being discussed in class. For instance, when the concept of alias is discussed, the students who study PASCAL or FORTRAN might be asked to investigate if aliases exist in these languages. They would explain their findings to the class in the next class period.

Week 4: The object-oriented programming paradigm, focusing on abstraction, inheritance, dynamic binding, and exception handling. Classes are handled in the same way as the procedural programming paradigm classes.

Weeks 5 & 6: The logic programming paradigm and PROLOG programming. The first in-class test is given in fifth week.

Labs are used when PROLOG is taught: the instructor first discusses syntax, then simple programming assignments are given on the whiteboard. The students compete by solving the problems on computers. Students who finish early are asked to help those students who might be having difficulties.

To check the progress of the students' work and to help the students write a better paper, they are required to turn in their first draft at the beginning of the sixth week. The papers are not graded; thus, they do not affect the students' final grade. The instructor, however, reviews the papers and passes comments back to the students. Along with the feedback, students also receive a separate problem assignment sheet. The problem sheet for each student contains problems that are related to the particular language they are studying. Students must address the newly assigned problems in the final draft and in the presentation. For example, students who study FORTRAN are asked to discuss the implementation mechanisms; characteristics of alias; variables and variable declarations; array operations; and subprogram parameter passing methods. Students who study EIFFEL are required to discuss its object-oriented programming features.

Students who study PERL are asked to describe the characteristics of associative arrays, major data types, and functions.

Week 7: Functional programming paradigm and Scheme programming.

Again, labs are used when Scheme is discussed. The labs are handled in a similar way to those of PROLOG.

Week 8: Concurrency and concurrency programming using JAVA; programming language syntax analysis, using attribute grammar notations.

Week 9: Peer evaluation of papers and the second in-class test.

At the beginning of the first class period in this week, papers are collected. The students are then divided into groups of three. Each group conducts peer evaluations of the papers. The students in each group are required to evaluate papers synchronously: all the students in the same group, except the author of the paper, grade the same paper during the same time period. It takes one class period for each group to evaluate one paper. During the evaluations, students are strongly encouraged to communicate with each other on the paper topics. The author of the paper being evaluated is asked to address questions from the other students. To quantify the evaluation results, each student is required to fill in one evaluation form for each paper he/she evaluates. The evaluation form is shown in Table 1. In addition to the student evaluation, the instructor reviews each paper individually and completes an evaluation form for each paper.

Table 1 The Paper Peer Evaluation Form

Author: Evaluator:	Criteria	Score/Comments
Part I Score	Visual Format (15 Points)	
	Grammar (5 Points)	
	Spelling (5 Points)	
	References (5 Points)	
	Discussions of the assigned problems (70 Points)	
Part II Comments	Describe three merits of the paper.	
	Explain three aspects of the paper that you think the author needs to improve.	

The evaluation forms for each paper are collected and passed to the corresponding author. The students are then given two choices: 1) if the students do not want to make any changes to

Table 2 Oral Presentation Rubric

Criteria	Excellent	Good	Fair
Content	<p>The presentation correctly and effectively addresses all the questions as required in the course syllabus and in the problem assignment sheet. It is easy for the audience to learn and understand the content. The presentation reflects a thorough study of the language.</p> <p>(60 Points)</p>	<p>The presentation correctly addresses all the questions as required in the course syllabus and in the problem assignment sheet. However, it needs some improvements in order for the audience to better learn and understand the content.</p> <p>(48 Points)</p>	<p>The presentation does not address all the questions as required in the course syllabus or in the problem assignment sheet. It makes it difficult for the audience to learn and understand. The speaker should spend more time studying the language.</p> <p>(36 Points)</p>
Visuals	<p>Slides greatly help audience to understand the presentation:</p> <ul style="list-style-type: none"> <li>• Important information is included and emphasized. The information is organized to maximize audience understanding.</li> <li>• Font is large enough to be seen by all.</li> </ul> <p>(10 Points)</p>	<p>Slides help audience understand the presentation:</p> <ul style="list-style-type: none"> <li>• Appropriate information is included.</li> <li>• Font size is appropriate for reading.</li> <li>• Some material is not supported by visual aids.</li> </ul> <p>(8 Points)</p>	<p>Slides little help audience understand the presentation:</p> <ul style="list-style-type: none"> <li>• Inappropriate information is included. Unimportant material is highlighted.</li> <li>• Font is too small to be easily seen.</li> </ul> <p>(6 Points)</p>
Attitude	<p>Speaker is confident. Speaker positively responds to questions. Speaker shows respect for the questioner.</p> <p>(15 Points)</p>	<p>Speaker shows comfort when interacting with audience.</p> <p>(12 Points)</p>	<p>Speaker reluctantly interacts with audience or avoids or discourages active audience participation.</p> <p>(9 Points)</p>
Handling of Questions	<p>Speaker consistently clarifies, restates, and responds to questions when needed. Responds to questions adequately and satisfactorily.</p> <p>(15 Points)</p>	<p>Most of the time, speaker qualifies, restates, and responds to questions when needed. Misses some opportunities for interaction. Responds to questions adequately and satisfactorily.</p> <p>(12 Points)</p>	<p>Speaker reluctantly interacts with audience Responds to questions inadequately.</p> <p>(9 Points)</p>

their papers, the average score from all the evaluation forms they receive will be the final score for their papers; 2) if students want to improve their papers based on the results in the evaluation forms, they must turn in the final version of the paper at the end of the tenth week. The instructor will calculate a new score for the paper if it has been updated according to the evaluation results. For example, if a student loses points due to spelling errors and later revises the paper to correct the errors, then the deducted points will be credited.

The second in-class test is given in the fourth class period of this week.

Week 10: Presentation.

Students are required to give a presentation in class, using Microsoft PowerPoint slides. Each presentation is followed by a 2-minute Question-and-Answer (Q&A) session. The students are expected to clearly address all problems as specified in the course syllabus and one or two problems as specified in the problem assignment sheet. When one student gives the presentation, the instructor and all the other students in class act as evaluators. Each evaluator scores the presentation according to the oral presentation rubric shown in Table 2. Some of the criteria in Table 2 are designed on the basis of the rubrics used for the oral presentation for senior design proposals in the Electrical & Computer Engineering and Computer Science Department at ONU.

For each presentation, the final score is determined by the average of the scores from all the evaluators.

### **3. Assessment of the Course Implementation**

Since the paper and presentation format was introduced into the OPL course several years ago, there is no comparable experimental data to demonstrate how well the paper and presentation format improves student learning. However, we have some positive observations from our teaching experiences:

Observation 1: The majority of students express a strong interest in learning when they have a chance to actively participate in the learning process itself. It is the case that the students are excited when they can teach others in front of the class. It seems that, sometimes, the communication among the students is more effective than the communication between the instructor and the students. During the discussion the students may use some terms or phrases that are more understandable to them.

Observation 2: The majority of students are enthusiastic about learning a language by themselves and writing a paper. Most students would like to spend more time improving their work. Although it is not required that the students must revise their paper after they receive the evaluation forms, most of the time all the students in class turn in their revised paper by the end of the quarter. This also demonstrates that the students have a desire to earn a better grade for the course.

Observation 3: From the instructor’s perspective, the majority of the students in class evaluate both the paper and the presentation in a responsible manner. Evaluations are generally objective and fair.

In a survey conducted at the end of the fall quarter of 2005, the students were asked to complete a questionnaire of eight questions. The results of the first six questions are listed in Table 3.

Table 3 Students’ Feedback to the First Six Questions in the Questionnaire  
(5 = Strongly Agree and 1 = Strongly Disagree)

Question	Mean Response
1. Do you like the paper and presentation format for the instruction of the OPL course?	4.00
2. Does writing a paper help you understand programming language concepts?	3.29
3. Does presentation help you understand programming language concepts?	3.71
4. Does the peer evaluation help you improve the quality of your paper?	4.43
5. Are the scores from the peer evaluations for your paper objective and fair?	4.57
6. Does the peer evaluation help you understand programming language concepts?	3.29

The responses to the questions 1, 4, and 5 confirm our observations 1, 2, and 3, respectively. The responses to the questions 2 and 6 indicate that students need more guidelines and help for writing and reading papers.

As to the question 7, “If a programming concept can be taught by the instructor in class or can be easily mastered from self-study on a particular language, which method do you prefer for the learning?”, 29% of the students prefer to be taught by the instructor; 29% prefer to study by themselves; and 42% prefer a combination of both. The results can be interpreted as follows: most students would prefer active learning to passive learning; however, students also want the active learning is to be guided by the instructor.

The question 8 asks the student, “How would the instructor improve the paper and presentation assignment?”, two responses are listed as follows:

1. “Longer period of self-study, followed by a longer paper and presentation. Make more of the quarter devoted to the research and the presentation.”
2. “need more guidelines.”

#### 4. Conclusions

In this paper, we described the implementation of the OPL course at ONU. In our implementation, we introduced the paper and presentation format. Generally speaking, students

are assigned to study a language that they are not familiar with, bring the knowledge back to the classroom, and share it with the other students in class. Students also write a paper on the language and then give a presentation to the entire class. The format is designed to encourage students' active involvement in teaching and learning activities. It has been shown that this approach is very successful: most students have shown a great interest in writing a paper on the language they have studied. Students reflect that their interest in the course topics is stimulated by conducting research on a language by themselves. From our instructors' perspective, the assignment increases students' activities in class, excites their interest in the course topics, and, overall, has a very positive effect on students' learning.

## References

1. Barrière, C., *Using the Web to Promote Active Learning Outside of Class Time*, 2002 ASEE Annual Conference and Exposition, Montréal, Quebec, Canada, June 16-19, 2002.
2. Bates, R. A. and Linse, A.R., *Preparing Future Engineering Faculty through Active Learning*, 2003 ASEE Annual Conference and Exposition, Nashville, Tennessee, USA, June 22-25, 2003.
3. Benek-Rivera, J. and Mathews, V.E., *Active Learning with Jeopardy: Students ask the questions*, *Journal of Management Education*, 28(1), 104, 2004.
4. Cook, E.D. and Hazelwood, A.C., *An Active Learning Strategy for the Classroom- "Who wants to win ... some mini chips ahoy?"*, *Journal of Accounting Education*, 20(4), 297, 2002.
5. Felder, R.M. and Brent, R., *Cooperative Learning in Technical Courses: Procedures, pitfalls, and payoffs*. Report to the National Science Foundation, 1994.
6. Felder, R.M. and Brent, R., *Effective Strategies for Cooperative Learning*, *Journal of Cooperation and Collaboration in College Teaching*, 10(2), 69-75, 2001.
7. Johnson, D.W., Johnson, R.T., and Smith, K.A., *Active Learning: Cooperation in the college classroom*, 2<sup>nd</sup> edition, Edina. MN: Interaction Book Co., 1998.
8. Miller, J.S., *Problem-based Learning in Organizational Behavior Class: Solving student's real problems*, *Journal of Management Education*, 28(5), 578-589, 2004
9. Millis, B.J. and Cottell, Jr. P.G., *Cooperative Learning for Higher Education Faculty*, Phoenix, AZ: Oryx Press, 1998.
10. Tebbe, P.A., *Brainstorming Exercises as an Active Learning Component of Thermal System Courses*, 2005 ASEE Annual Conference and Exposition, Portland, Oregon, June 12-15, 2005.