

Fostering Collaboration in Teaching Computer Programming through Sharing of Relevant Examples: a Proposed Project

Peter Y. Wu

wu@mail.rmu.edu

Department of Computer & Information Systems
Robert Morris University
Moon Township, Pennsylvania, USA

ABSTRACT

While we teach with examples, students learn from doing. Each example should illustrate a specific concept, and each assignment puts the concepts in an application for exercise. The IS curriculum is particularly strong in the emphasis of application in teaching programming. Good examples are difficult to find, and substantial effort is necessary to design and develop them. A good teacher maintains a reserve pool of illustrative examples and interesting exercises, and knows what each one is effective for. While we can easily gather a good collection of many examples, we need a good search tool to find the right one. Our need in each teaching situation can be very specific. We need to leverage a good sharing facility to foster collaboration. Good teachers who use the system may also want to contribute to the system. It should also keep track of contribution credits and usage statistics to provide additional information for teaching performance review. For a good search tool, we need to work on categorization of the content and design the user interface for easy browsing, searching, and making contribution.

Keywords: computer programming, teaching examples, categorization, dependencies

1. INTRODUCTION

The IS curriculum offers several courses in the programming track. Even when most IS graduates do not eventually become computer programmers or software engineers, the training in programming is fundamental to the understanding of computers and software. An appropriate fluency and ability to reason about software components remain essential for a system analyst, as well as many other IS/IT support or management roles. The growing use object-oriented programming technologies and object-oriented approaches in IS/IT practices in the past decade has already made the need to understand software more acute among IS/IT professionals (Caputo 2004). Despite the fact that some of our colleagues promoting MIS may advocate minimizing teaching of com-

puter programming, it remains largely an essential part of the core in the IS curriculum (IS 2002, SIGCSE 2001). Effective teaching of computer programming is important to IS education and the focus of this paper.

Good teaching needs good examples. Each example should illustrate a specific concept. Good teachers know that good examples are difficult to come by. It is difficult not because of the scarcity of examples – indeed there are plenty in the plethora of textbooks (Deitel 2007, Malik 2006, Schneider 2007), but we often have very specific needs and desires when we intend to illustrate a specific concept. Consider the concept of arrays and the use of loops in programming. The two are distinct concepts but also closely related. Depending on the order in

which we introduce the two different concepts, we may need examples to illustrate one without bringing in the other. On the other hand, students learn from doing – working on exercises designed for them to put into practice certain concepts. We will want to avoid topics not covered yet, but also very much want to build on those already covered. More so is the case when we need to design test or examination questions for learning outcome assessment.

Furthermore, we focus on the value of applications in IS education. It is always desirable to have examples and exercises in areas of application students are interested in and, even better, already knowledgeable about, while avoiding the need to cover specific domain knowledge when it is not relevant. Using a loop to compute mortgage amortization should be interesting to students in finance, but engineering students may feel that it is irrelevant to learn about mortgage calculations.

Good teaching examples are difficult to come by, given these specific needs and desires in the different varieties of teaching situations. It is particularly important in the IS curriculum, since want to train up students to be prepared for a variety of application areas. We believe it will be very beneficial for the teachers of the programming courses to collaborate and share information. Our effort in collaboration however is limited to personal discussion, face-to-face, by phone and/or e-mail. We can, and do, share information. But we can only achieve that handling one piece at a time. We need a better facility to share good illustrative examples and interesting assignments and exercises, including test and exam questions.

We propose to pool together various programming examples, assignments as well as exam questions from the current teachers, along with appropriate indication of the specific concepts each is built on, or intended to illustrate or exercise. The syllabus of each programming course provides a fairly thorough description of the topics to cover. That will also be the basis for us to categorize the sample items of programming examples, assignments, and exam questions. We then need to sort out the dependencies of the various concepts and topics involved. To facilitate sharing and foster collaboration, we propose to build such a library system for

the teachers. We will need to design appropriate user interfaces for easy browsing, searching, as well as making a contribution to the collection of materials. The system should record the contribution credits, and keep track of usage statistics. Perhaps on an annual basis or on demand, the system may also report the contribution credits for teaching performance review. This paper briefly explains how such an example or exercise may be properly categorized, and describes in the manner of a project proposal how we intend to design and build such a system.

2. CATEGORIZING A TEACHING EXAMPLE AND/OR EXERCISE

Consider the case of C++ programming using for loops. We want to illustrate the use of various control variables in the loop, but the students must also understand the concepts of local and global scopes involved. An illustrative example may involve the following cases:

```
int n = 0;
for (n = 1; n < 3; n = n+1)
{
    cout << "LOOP:" << n << endl;
}
cout << "FINAL:" << n << endl;
```

Output:
LOOP: 1
LOOP: 2
FINAL: 3

Figure 1: Loop illustration – Case 1

```
int n = 0;
for (int n = 1; n < 3; n = n+1)
{
    cout << "LOOP:" << n << endl;
}
cout << "FINAL:" << n << endl;
```

Output:
LOOP: 1
LOOP: 2
FINAL: 0

Figure 2: Loop illustration – Case 2

```

int n = 0;
for (n = 1; n < 3; n = n+1)
{
    n = 3;
    cout << "LOOP:" << n << endl;
}
cout << "FINAL:" << n << endl;

```

Output: **LOOP: 3**
 FINAL: 4

Figure 3: Loop illustration – Case 3

The simple exercise is very illustrative, but it requires a reasonable understanding of local and global scopes of variables. Yet variable scope is in reality a different though related topic, and beyond local and global scopes, while the topic of variable scopes can go much deeper. We do recognize these topics and sub-topics to have dependencies on one another. Further, each topic, or sub-topic, needs to be uniquely identified in refined details. Our proposed project would first require us to sort out these topics from the course syllabus, and layout the dependencies among them. Then, for each illustrative example, such as the one show above in 3 cases, need to be properly categorized as intended to illustrate control variables of the loop construct, but dependent on the understanding of local scopes. Furthermore, it assumes the knowledge of using simple variables and data types, along with simple input/output statements. Note also that one can easily modify such an example into test questions for outcome assessment.

When we have the detailed map of the topic dependencies, we can appropriately categorize each example to indicate:

- the topic the example is designed to illustrate, and
- the topics the example requires the student to know reasonably well already.

The rest of the topics we can then assume they are not relevant yet in this example, and are probably something for the future. In our proposed project, we will then collect examples and have each example properly categorized for effective searching as well as browsing.

3. TWO PHASES OF THE PROJECT

We propose a research project to develop such a system in two phases.

In the first phase of the proposed project, we should first decide on the programming language for this work, and sort out the refined topics of coverage to layout the dependency graph of the topics to be covered. Then, we will focus on gathering content. We will gather and analyze existing materials of programming examples, assignments, and exam questions from the faculty members of the teaching staff, as well as other willing souls among ISECON colleagues. The main work in this phase, however, is the detailed design of user interfaces to facilitate browsing, searching, and making contribution, along with how the system would be used. In the design of the user interface, we can spell out the design requirements. We will need to design software development tasks suitable for student projects and bring in student aid for the development effort. Many of these projects should be appropriate for a current student with the right skills and preparation to take up for pay or for credit. The goal of this phase is a system that is reasonably robust, ready for initial use.

For the second phase of the proposed project, we plan to begin with gathering feedback from users. We will need to fix bugs, and carry out the development work for system enhancement. That will be on-going, and should aim at a second release of a much more usable and robust system. It is then also time to write good on-line documentation such as user guide, and the development of promotional materials and activities to encourage use of the system. Promotion to use the system and on-going sustenance are the goals of this phase.

4. POTENTIAL IMPACT OF THE PROPOSED PROJECT

A large part of our effort in teaching the programming courses is in the search of good examples and interesting assignments. It means a substantial effort for any teacher to design and develop these materials from scratch. When the proposed web-based system becomes available and there is a sufficiently large collection of teaching examples, its immediate use should mean a

tremendous saving in time and effort for the teachers.

Teachers usually take up ideas of examples and assignments and modify them, adjusting to appropriate use for the courses they are teaching. The work involved quite often leads to other new ideas, of even better examples or more interesting assignments. The proposed system will provide an easy way for a teacher to make a contribution, thus allowing others to use his/her ideas. This is important for the growth of the resource pool in the system, to make its content even richer. The system will record the individual and the time of each contribution to give proper credit, and keep track of usage statistics as well as comment and feedback from other users. It would then provide a forum for discussion, leading to further collaboration. Periodically, we may also obtain reports on contribution credits to be relevant information for teaching performance review.

The most important impact expected in long term is the improved quality of teaching. Students in programming courses often express that they learn from examples and become skillful and confident through working on programming assignments. A rich pool of examples, assignments and exam questions will be a very valuable resource for the teaching. A facility to allow easy sharing will undoubtedly improve our teaching. The impact should become more obvious in long term.

While we already have a reasonable size volume of programming examples, assignments, and exam questions, it is still important for the pool to continue to grow. We also hope that when we gather a very large pool, it will become the source of materials for the development of self-paced distance learning courses in the BSIS programming track.

5. FUTURE SUSTENANCE

The future sustenance of the proposed project will rely on two major factors. The teaching staff must continue to use the system to share content and to make contributions. The other factor is the technical maintenance of the system to keep it up-to-date and easy to use.

It is important to have sufficient materials to get the system started for use, and the system has to be user-friendly. We should be able easily gather enough materials from just a few in our teaching staff in support of the project. It is therefore most important to design and develop a system that is truly easy to use.

A good robust system will be relatively easy to maintain. We will need to keep up with the changing desires of the user community, and continue with enhancements to support the changing world of information technologies. The proposed project will need a faculty member technically competent to take ownership of the system. In fact, we expect the future enhancements to make interesting independent study projects. Students interested in taking up these projects will get academic credit for their effort in programming support.

When the system is easy to use, and it brings a good benefit, it will be used. We believe a good teacher will always find it rewarding when his/her ideas are adopted for use, and will find it interesting to discuss these ideas about drawbacks and improvements. But to give credit where it is due, it is only fair for the system to properly credit those individuals who make good contributions. The system will record all contributions and their credits, and keep track of the usage statistics along with comments and feedback. Perhaps annually, or on demand, the system should duly produce reports on contribution credits. Faculty members who made contributions can include these reports in their teaching dossier for performance review. When proper credit is given, the teachers will be encouraged to continue their use of the system, and make contributions.

6. SUCCESS CRITERIA

Use of the proposed system would already mark success for the proposed project. Users only use the system when it is easy to use, and it brings a good benefit to them. The system has to be used, and the feedback from the user community, that is, our own teaching staff of the programming track, will indicate the level of success. When all the teachers in the IS programming track would use the system, we will claim initial success. The continued use is still necessary for success in the coming future.

Finally, it is when the students, who take the programming track courses offered in the IS curriculum, find them excited in learning programming because of the use of good illustrative examples and the interesting programming assignments. Then the proposed project is successful.

7. REFERENCES

- Caputo, Kovacs, and Turcek (2004). Defining the Essential Skill and Functional Areas of Study in IT as Measured by a Survey of Field Professionals. *Proceedings of ISECON 2004*, Newport, RI., v.21, SS.2215, October 2004.
- Deitel and Deitel (2007), *Java How to Program*, published by Prentice, 2007.
- IS2002 (2002) Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems 2002, curriculum effort of Association for Computing Machinery (ACM), Association for Information Systems (AIS), Association of Information Technology Professionals (AITP).
- Malik (2006), *C++ Programming: From Problem Analysis to Program Design*, published by Thomson Course Technology, 2006.
- Schneider (2007), *An Introduction to Programming using Visual Basic 2005*, published by Prentice Hall, 2007.
- SIGCSE (2001). *CS Body of Knowledge - Information Management Computing Curricula 2001*, Association of Computing Machinery, 2001.