



# An Interactive Web-Based Tool For Teaching Computer Programming

Amal Abdel-Raouf

Computer Science Department, Southern Connecticut State University, USA  
Computers and Systems Department, Electronics Research Institute, Giza, Egypt  
abdelraoufa1@southernct.edu

## ABSTRACT

Lab time dedicated to programming practice in most schools is typically not sufficient for students to succeed as professional programmers. In this paper, we introduce an interactive tool for teaching computer programming. The tool can be used outside the classroom as it is web-based, can be easily integrated with any management learning environment such as BlackBoard, and provides better opportunities for distance learning and online education programs. The tool engages students as active learners and considers student learning styles to effectively improve the learning process. Our experimental study shows that using our tool significantly increased the achievement percentage of our class learning outcomes.

## Keywords

Interactive learning tool, Management learning environment, Student motivation, Learning styles, Learning outcomes.

## 1. Introduction

Teaching computer programming is a challenge as students not only need to comprehend the theoretical concepts but must also practice programming using software programming languages. Most schools dedicate limited lab time for computer programming practice. At the same time, labs are not easy to implement in an e-learning environment, as they require an available instructor to provide continuous feedback to students on their work. Students tend to make many errors when they start to learn programming; especially problematic are compilation errors because they are not easily understood or fixed. This creates a barrier between students and programming, which in turn can cause them to drop the class and hurts the students' retention rate.

In the literature, many web-based learning tools have been introduced to help students understand various subjects. Cheng in (Cheng, 1997) introduced e-learning as a breakthrough in education. Khan in (Khan, 1997) created an instructional guide for web based tutoring tools. Afterwards, many web-based tools were implemented with the advantage of including multimedia components to engage students. However, these tools were not adaptive according to the definition of Brusilovsky in (Brusilovsky, 2003). The first adaptive electronic textbook was introduced in (Brusilovsky et al, 1998).

In (Lee et al, 2004 ) all multimedia instructions needed for web-based training or team-learning events are listed; these instructions are very similar to what is required during computer programming tutoring. Many web-based learning tools have been developed with the aim of increasing student motivation and making learning fun (Dagger et al., 2007; Wolf, 2003). Specifically for computer programming learning, some simulators were implemented to help students practice programming (Simulators, 2007). Other tools were introduced such as (Abdel-Raouf, 2014) targeting online teaching, and (Fergany et al., 2001) for teaching the C programming language. The design of the tool discussed in this paper was proposed in (Fergany and Abdel-Raouf, 2011). However, this tool was only described and not developed, so there was no way to evaluate its effectiveness.

Through our interactive learning tool, students are able to study the computer programming concepts, test their knowledge, and practice designing the solution before they start the implementation phase. Additionally, the tool provides the instructor with a way to communicate with students, monitor their progress and identify weaknesses and learning styles. These features allow the instructor to adjust teaching methods to match the students' needs.

The rest of the paper is organized as follows: the second section describes the features of the web-based tool and lists each feature's advantages. Section three details the architecture of the learning tool and the different modules it includes. Section four presents our experimental results and discussion. Finally, the conclusion is presented in section five.

## 2. Web-based Learning Tool

The different factors affecting the motivation of university undergraduate students taking computer-programming courses are listed in (Law et al, 2010). The most remarkably motivating factors are: individual attitude and expectation, clear directions, and reward and recognition. In our tool, we considered all of these factors in addition to achieving the learning outcomes of the computer programming course. The following subsections list the features of our tool and demonstrate how they positively affect student learning.

### 2.1. Learning Styles Support

Students play a major role in the learning process. One of the important objectives of this tool is to engage students as active learners. As students tend to learn in different styles, our tool is built to be diverse enough to accommodate all students (Liaw et al., 2007).

A navigation menu is used to facilitate browsing different computer programming concepts and provide students with increased control over the education process. The goal is to create for each student a learning experience that is dynamic, interactive and individually customized according to her learning style (Ng and Marriott, 1996 ).

While motivating students for customized navigation, the tool also elucidates a coherent association of previous, current, and upcoming concepts and provides avenues for understanding relationships between concepts as described in the next subsection.

## 2.2. Navigation Guidance

A major challenge in developing computer programming education tools is providing students with the guidance to help them better understand the concepts and gain the computer programming skills. We recommend that students start with the concepts and background section, and then continue to learn concepts in the sequence of the learning outcomes as listed in the textbook. This is appropriate for sequential learners, but other students may try to learn a specific concepts that is needed for an assignment or in order to expedite the learning process.

A tree structure is used to hold the relationships among the different computer programming concepts. Each concept is represented as a node in the tree. A concept which is a prerequisite for another concept is represented as its parent node. In case a student runs into difficulties, the tool uses the tree structure to refer the student to the prerequisite concepts that need to be learnt first.

## 3. Web-based Learning Tool Architecture

In this section, we introduce the architecture of our web-based learning tool. The tool is designed as a typical layered architecture system as described in (Sommerville, 2010). The system is organized into layers with related functionality associated with each layer. Each layer provides services to the layer above it so the lowest-level layers represent core services that are likely to be used throughout the entire system. Therefore, this model is generic and can be used for teaching any programming language by updating the lowest-level layer. The different layers of the tool's architecture, as shown in figure 1, are described in the following sub-sections.

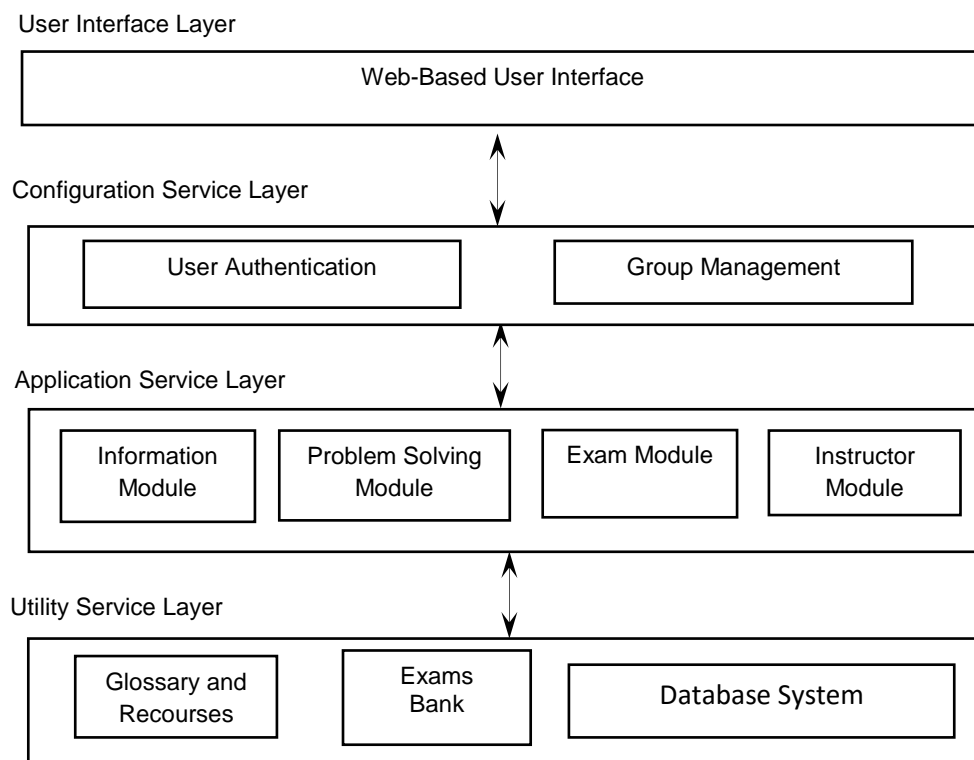


Figure 1: Web-Based Learning Tool Architecture

### 3.1. User Interface Layer

The tool is interactive so that the student plays a crucial role in the learning process. Through the user interface layer, the tool allows the students to test their knowledge and practice exam questions. The student is able to decide the level of difficulty, repeat some modules with different difficulty levels, and check the Need-Help icon for difficult modules when she needs extra help. Moreover, the student is able to submit questions or comments to her instructor and receive the instructor's response.

### 3.2. Configuration Service Layer

This layer is used for service configuration. It is responsible for user interface management, user authentication authorization and group management. The user authentication module makes sure that only students with permission can access the tool. Every student is given an account, an ID and a password to access the account. The group management module allows the instructor to create teams of students and give permissions to teams to access specific information.



This allows teamwork, so every team can access the information related to their own work without interference with other teams in the class. Another example of this module's application is a student who misses an exam can be given access to the exam later while the other students cannot access the exam again.

### **3.3. Application Service Layer**

This layer includes all the applications that provide service to the students who access the tool as described in the following sub-sections.

#### **3.3.1. Information Module**

Students use this module to understand the fundamentals computer programming concepts. The module includes the needed background and a brief description of each concept. The module also uses the tree structure to enforce the relationship between the current concept and any related one. Moreover, the module includes the syntax information and all the rules to apply for the specific computer programming language used. The information module also includes, for each concept, programming tips and common errors to avoid. Additionally, students have the opportunity to review material at any time.

#### **3.3.2. Problem-Solving Module**

This module represents the work space where students can apply the knowledge they gained for each concept through the information module above. It gives students an opportunity to practice solving real-life problems using the programming language and requires students to make decisions about data types and operations. Students learn important problem solving techniques such as breaking down large problems to solve them then integrating the resultant solutions. Each problem in this module is intended to reinforce some programming language concept and/or a problem solving technique. This module also helps students in an interactive mode to implement and test their problem solutions. The system then reports, to the students and the instructor, the concepts and the techniques that need more study.

#### **3.3.3. Exam Module**

Students access this module to test the knowledge they gained from the information module and after practicing in the problem solving module. This module includes different levels of quizzes and exam questions. It always starts with the easiest level, usually multiple-choice questions to help students recall the important points of each concept as well as to review prerequisite material. Then the module moves to the next level, providing some coding question. All exams/quizzes include time limits according to their level of difficulty. When the exam/quiz ends, students have the opportunity to review the incorrect answers as the module brings up text fields on the screen, explaining why the answer is incorrect. When the exam ends, the testing module displays the student's grade together with some recommendation to review specific learning concepts.

#### **3.3.4. Instructor Module**

The instructor as a user has more privileges than students. The instructor uses this module to monitor students' performance. Important performance details include how many times a student accesses the tool, the date of the last visit, the problems the students solved and if the student had any difficulties solving the problems. As the instructor has writing access to the learning materials, the instructor uses this module to add, delete, or edit different course materials according to the students' need. This module enables the instructor to evaluate all submitted work and respond with comments. Through the different activities of this module, the instructor can diagnose student problems, investigate and examine ways to help students, and change the teaching methods accordingly.

### **3.4. Utility Service Layer**

This layer is responsible for system support and storage as described below.

#### **3.4.1. Exams Bank**

All exam and quiz questions are saved in the exam bank, allowing the instructor to change exams and quizzes every term. Moreover, the student account saves the topics studied, tests taken and the grades earned in each test.

#### **3.4.2. Glossary and Reference**

Two different glossaries are included in his module. The first glossary includes the definitions together with the key terms of the computer programming language presented in the course. The second glossary includes the used programming language terms, operators, API's and standard libraries. The second glossary also includes a list of the language reserved words, primitive types, their sizes, etc. In our case we used Java (Gaddis, 2015) as the core language for this tool but any other programming language can be used by exchanging the glossies and references with those corresponding from the alternative language of choice.

#### **3.4.3. Database System**

The database system saves all data needed for the learning tool and holds all related information. The database is used to store all assignments, course materials, notes, textbooks, development environment, and any other information needed for the computer programming course. For each student, the database holds an account with the student's information: name, ID, contact information, grades, term, etc. Additionally, the database system contains progress reports for each student

enrolled in a course. The database size is determined according to the number of students taking the course. Therefore the database size should be increased when the student number increases.

## 4. Experimental Results

In order to evaluate our tool we followed the evaluative technique described in (Chumley-Jones et al., 2002). We conducted our experiment on two control groups. The first group is a group of 34 students in the introductory computer-programming course at Southern Connecticut State University, where the web-based tool was adopted. The results are compared with the results of the second group of 32 students who were not taught using the web-based tool, but otherwise experienced the same learning environment; same instructor, textbook, class room, computer lab, etc.

### 4.1. Evaluation Criteria

Our evaluation criterion is the percentage achievement of the course learning outcomes. According to (Rauhvargers et al., 2009), the learning outcomes are defined as 'statements of what the learner will know, understand and be able to demonstrate after completion of a program of learning'. The learning outcomes (LOs) that are considered for the computer-programming course are given in table 1.

**Table 1: Learning Outcomes of the Computer Programming Course**

LO1	Ability to declare and assign variables of different data types
LO2	Ability to use appropriate tools to edit, compile and run Java programs
LO3	Ability to incorporate control statements
LO4	Ability to incorporate looping structures
LO5	Understanding of the abstractions of a method and parameter passing
LO6	Ability to incorporate one-dimensional arrays

In (Peters, 2004), the author described two techniques for learning outcomes assessment: summative or formative. The summative technique is based on quantitative values that indicate what the students can accomplish using what they learn; hence score, grade, marks or ranking can be used as measured values. The formative technique is based on the feedback given to students to improve the learning process.

In our experiment, we used the summative technique and for each learning outcome; an assessment, either an assignment, quiz or exam question, was designed to measure the achievement of the learning outcome. For each learning outcome, the average assessment score is calculated for both groups of students and the results are shown in figure 2. Group1 represents students who used the web-based tool while group 2 represents students who did not use the web-based tool.

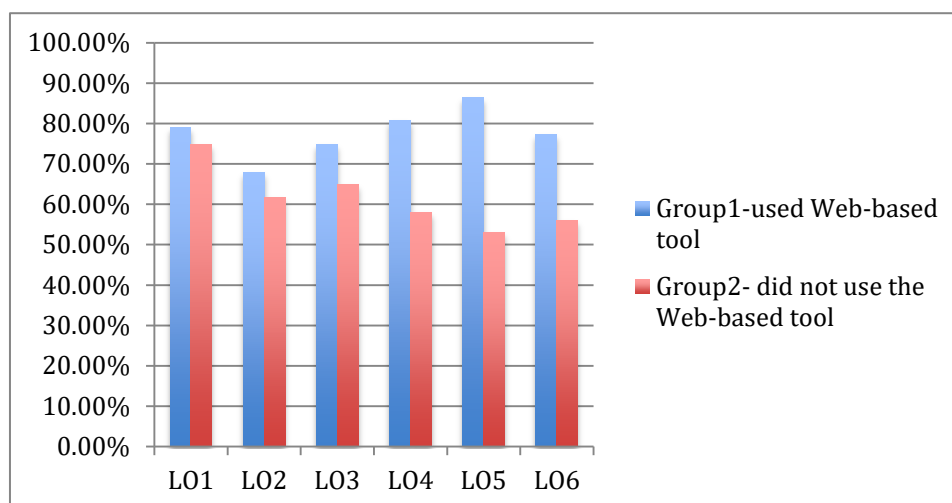


Figure 2: Comparison of the Percentage of Achieving Learning Outcomes

### 4.2. Discussion and Analysis

The use of the web-based learning tool resulted in a percentage of improvement in achieving all learning outcomes as shown in figure 3. The results indicate that the tool was well received by the students.

The improvement percentage was minor for the first three learning outcomes as the concepts tend to be easy at the beginning of the semester and the achievement level was satisfying without using the tool. However as the learning concepts become more advanced, the improvement percentages increase significantly. LO5, abstractions of a method and parameter passing, is a challenging concept for students to understand and using the tool improved its achievement by almost 63%. Furthermore, the improvement percentages in LO4 and LO6, loops and arrays respectively, are nearly the same. This is likely as a result of the relationship between the two concepts, as students usually use loops when they access arrays in a computer program. Consequently, the achievement increase in LO4 using the learning tool translates to a similar achievement increase in LO6.

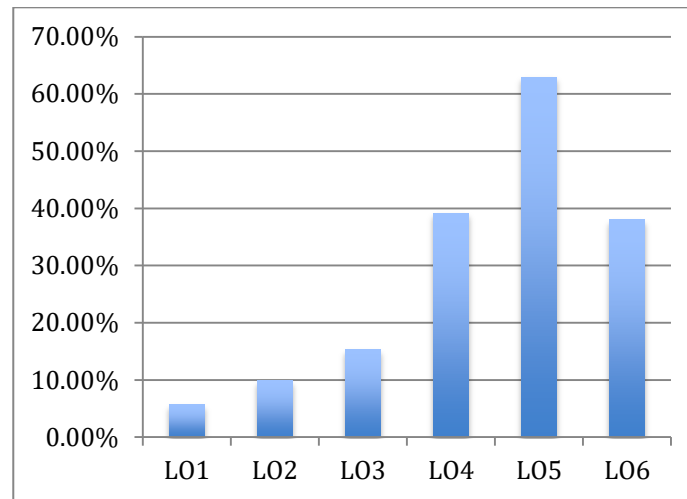


Figure 3: Improvement percentage of achieving learning outcomes by using the web-based tool

## 5. Conclusion

In this paper, we introduced a web-based learning tool to teach students with different background and various learning styles computer programming concepts. At any time, the tool acts as an available instructor to answer students' questions and reinforce the connections between the different topics. Using this interactive tool makes students more comfortable practicing programming during the learning phase, helping to improve student engagement, and allows instructors to better serve their students by updating their teaching methods based on student information provided by the tool. The experimental results show that using the tool has effectively improved the achievement of course learning outcomes in an undergraduate computer science course.

## REFERENCES

- 1) Abdel-Raouf, A., 2014. *An Integrated E-learning Environment for Teaching Computer Programming*, in the *Proceedings of the 2nd Conference on E-Learning in the Arab World (EELU)*, pp. 239-251, Cairo, Egypt.
- 2) Brusilovsky, P., Eklund, J. and Schwarz, E., 1998. *Web-based education for all: a tool for development adaptive courseware*. *Computer networks and ISDN systems*, 30(1), pp.291-300.
- 3) Brusilovsky, P. and Peylo, C., 2003. *Adaptive and intelligent web-based educational systems*. *International Journal of Artificial Intelligence in Education*. IJAIED 13, 159-172.
- 4) Cheng, I.F., 1997. *Internet Interactive Authoring, Tech and learning- a breakthrough for education*, *Proceedings of the COMPSAC'97-21st International Computer Software and Applications Conference*.
- 5) Chumley-Jones, H.S., Dobbie, A. and Alford, C.L., 2002. *Web-based learning: Sound educational method or hype? A review of the evaluation literature*. *Academic medicine*, 77(10), pp.S86-S93.
- 6) Dagger, D., O'Connor, A., Lawless, S., Walsh, E., & Wade, V. P., 2007. *Service-oriented e-learning platforms: From monolithic systems to flexible services*. *Internet Computing, IEEE*, 11(3), 28-35.
- 7) Fergany, T., Eggert, D., and Fischer, A., 2001. *A Web-Based Learning Environment for the C-programming language*, *The IEEE Educational Software Workshop*, Cairo.
- 8) Fergany, T., and Abdel-Raouf, A., 2011. *An Active Learning Environment for Teaching Object-Oriented Concepts, Design and Implementation*, in the *Proceedings of the ASEE Northern Section Conference*, Hartford University, Hartford, CT.
- 9) Gaddis, T., 2015. *Starting out with Java from control structures through objects*, 6th Edition, Addison-Wesley Publishers Ltd, New York.
- 10) Khan, B.H. ed., 1997. *Web-based instruction*. *Educational Technology*.
- 11) Law, K. M., Lee, V., & Yu, Y. T. , 2010. *Learning motivation in e-learning facilitated computer programming courses*. *Computers & Education*, 55(1), 218-228.
- 12) Lee, W.W. and Owens, D.L., 2004. *Multimedia-based instructional design: computer-based training, web-based training, distance broadcast training, performance-based solutions*. John Wiley & Sons.
- 13) Liaw, S. S., Huang, H. M., & Chen, G. D., 2007. *Surveying instructor and learner attitudes toward e-learning*. *Computers & Education*, 49(4), 1066-1080.



- 14) Ng, J., and Marriott, A., 1996. A survey of Users of a Web-bases computer graphics course, IEEE Computer Graphics and Applications, Vol. 16, NO 3.
- 15) Peters, J., 2004. Learning outcomes and information literacy. SCONUL.
- 16) Rauhvargers, A., Deane, C., and W. Pauwel. 2009. Report from working groups appointed by the Bologna Follow-up Group to the Ministerial Conference, April28-29, in Leuven/Louvain-la-Neuve.
- 17) Simulators for Teaching Computer Science, 2007. <http://vip.cs.utsa.edu/simulators/> (accessed 04.08.2016).
- 18) Sommerville, I., 2010. Software Engineering, 9th Edition, Addison-Wesley Publishers Ltd, New York.
- 19) Wolf, C., 2003. IWeaver: towards 'learning style'-based e-learning in computer science education, In Proceedings of the fifth Australasian conference on Computing education-Volume 20 (pp. 273-279).