COMPARISON OF ONLINE TOOLS THAT CAN BE USED TO ENHANCE CS1 DISTANCE LEARNING

Mark S. Hall
Associate Professor of Computer Science
Computer Science, Engineering, Physics & Astronomy (CSEPA)
University of Wisconsin – Marathon County (UWMC)
University of Wisconsin - Colleges
Wausau, WI 54401
mark.hall@uwc.edu

Abstract

Each of the distance education campuses has between 1-5 students enrolled in the lecture component of the CS1 course. These students face isolation fears due to low number of students available to discuss problems and to an instructor who is not on the same campus. Another major problem is that the students enrolling into the UW–Colleges are that many students do not have the necessary math requirements to enroll in any CS course. They are forced to take remedial math courses prior to actually taking their first CS course. It is easy to understand why these students struggle with their problem solving skills as they failed to take the necessary college prep courses while in high school. To succeed, these students need to be exposed to as many opportunities to solve problems, either in homework assignments or programming assignments. However, the disadvantage is that this requires intense effort on any instructor to grade these assignments. Students take graded assignments more seriously than if they are not graded.

The remote distances between students and students, and also with the instructor create additional problems. Using group assignments or peer-programming can be beneficial for at-risk students to increase their success rate. However, the remote distances between students make this near impossible. In face-to-face courses, it can be hard to get students to show up for office hours so distance education compounds that problem.

1 Introduction – Academic Environment

Many journal articles and papers have documented the problems that are facing the Computer Science discipline in the first two years of the curriculum. The core Computer Science courses usually are programming classes with student drop-out rates higher than the discipline would like to see. Many new pedagogically ideas are constantly being offered as aids to combat this problem.

1.1 Background

The thirteen 2-year University of Wisconsin (UW) Colleges serve students over the entire state of Wisconsin [1]. The Computer Science (CS) program in the UW Colleges is designed to be easily transferable to the 4-year institutions within the UW System. The nationwide decline in CS enrollments reported in the 2003-2004 Taulbee Survey [2] has affected the UW System, and has been particularly hard on the 2-year UW – Colleges (UWC), which are relatively small colleges, with a combined enrollment of approximately 12,000 students. In addition, the majority of our students are in the bottom 60% of their high school rank. The majority of our students enroll to prove that they can achieve at the University level in order to be accepted at a 4-year University.

1.2 Enrollment Issues

A trend of declining enrollment in Computer Science (CS) is being felt at the 2-year University of Wisconsin (UW) Colleges, as is the case nationally. UWMC is the third largest two-year campus in the UW Colleges [3]. It is safe to say that those are NOT the numbers that were active in the classroom at the end of the semester. The loss of these potential students for the core Computer Science courses are very troubling. Many research articles have documented that students struggle with learning the syntax of programming languages. Something needed to be done to recruit and then retain these students.

1.3 Problems

A major problem is that the students enrolling into the UW–Colleges are that many students do not have the necessary math requirements to enroll in any CS course. They are forced to take remedial math courses prior to actually taking their first CS course. It is easy to understand why these students struggle with their problem solving skills as they failed to take the necessary college prep courses while in high school. To succeed, these students need to be exposed to as many opportunities to solve problems, either in homework assignments or programming assignments. However, the disadvantage is that this requires intense effort on any instructor to grade these assignments. Students take graded assignments more seriously than if

they are not graded.

1.4 Designing Algorithms and Critical Thinking Skills

As I looked at the guidelines established by the UW Colleges for teaching the general concepts for this course, the emphasis that course coverage for designing algorithms should take at least a minimum of six hours. And the reality is that every programming assignment typically forces students to design a totally different algorithm. In order to be successful in this course, for the students and the professor, somehow, someway, the critical thinking skills of students have to be improved. Improvements in critical thinking along with the understanding of the basic control structures will allow students to be successful writing simple programs using any language. It is obvious that more time on task can be very helpful for developing these critical thinking skills. However, more assignments that need to be graded so feedback can be given to the student can overwhelm any instructor.

2 Benefits of Online Tools

The rationale behind find online tools is to provide extra benefits to the student, the instructor, and hopefully, to the department.

2.1 Benefits to the Student

The potential benefits to the students are: [4]

1. Higher Test Scores
2. Reduction or elimination of student frustration (hitting the “brick wall”)
3. More Efficient Studying
4. Better Grades on Projects

2.2 Benefits to the instructor

The potential benefits to the instructor are: [4]

1. More time spent n concepts than syntax and grammar
2. Grading relief / Automatic class rostering
3. Frees up office hours
4. Decrease attrition – increase retention

2.3 Benefits to the CS Department

The potential benefits to the CS Department are: [4]

1. Retention of students to increase advanced course enrollment
2. Language Switching
3. Confidence of Achievement Level
4. Automated Placement Exam

If more students are successful in the CS1 and CS2 courses, then enrollment in the upper division computer courses will also increase. Typically, students are asked to switch languages as they take upper level courses so these tools will allow students practice with the new language. Instructors of advanced courses can have a sense that students have gained mastery over key computer science concepts. Universities can use these tools to test mastery of key concepts to see whether High School students should be allowed to skip the introductory computer science course.

3 Online Tools Comparison

It was obvious that something had to change in my classroom in order to decrease the drop rate and to help retain these students for future core Computer Science courses. I started investigating whether there were any online Computer Science tools that could I could use in my courses that would help students in any way to feel more comfortable and confident when writing algorithms to solve problems.

3.1 CodeLab

CodeLab is the web-based interactive programming exercise system for introductory programming classes in Python, Java, C++, C and other languages. The tool was designed to reduce attrition and raise the overall knowledge level of the students in the courses. It is a heavily used, seasoned system that has been used in over 100 institutions. Since 2002, CodeLab has analyzed over twenty-two million exercise submissions from more than 75,000 students. [4]

3.1.1 Pedagogy

The pedagogy behind CodeLab mimics techniques used widely in other subjects, such as mathematics and foreign language study. The main idea of the pedagogy is to provide large numbers of self-paced, highly interactive exercises that focus on key ideas of programming. These exercises are intended to augment, rather than to replace, the traditional "whole program" assignments in the first year of undergraduate study. CodeLab is a web-based tool that enables faculty to assign exercises to students and monitor student progress. For the student, CodeLab

provides experience with fundamental elements of syntax, semantics, and basic usage of the programming language. The tool provides immediate feedback on correctness and often offers suggestions for fixing errors. Students can proceed at their own pace, subject to deadlines imposed at the instructor's discretion.

CodeLab has over 300 short exercises with each exercise focusing on a particular programming concept. The GUI is shown in Figure 1. The exercise is presented to the student and the student types in their solution to the exercise. A typical homework exercise is shown in Figure 2. The system immediately judges the correctness of the solution, and even offers hints when the submission is incorrect. If incorrect, the student modifies their solution, and submits it for verification. "Through this process, the student gains mastery over the semantics, syntax and common usage of the language elements."[5]

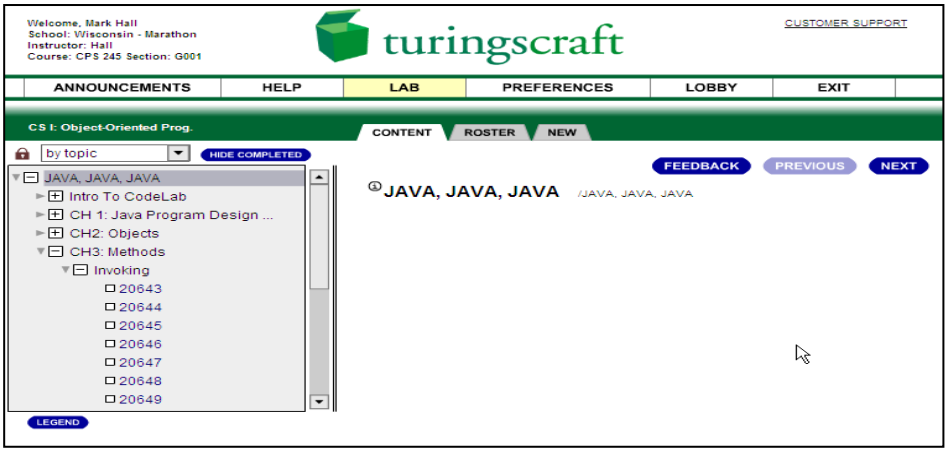

Figure 1: CodeLab GUI

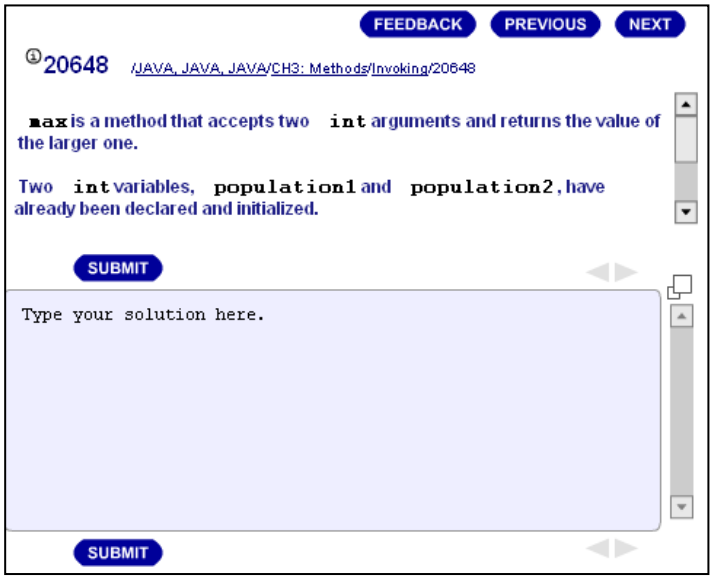

Figure 2: CodeLab Exercise

3.1.2 History

Turing's Craft was founded in 1999 by David Arnow and Gerald Weiss. They are both professors of Computer and Information Science at the City University of New York. To aid in teaching Computer Science concepts, "Arnow developed the WebToTeach system to address the limited opportunities for computer science students to practice the concepts taught in the classroom." [5] The early academic versions of WebToTeach were used by thousands of students in several colleges and high schools since 1990. A National Science Foundation grant was awarded to them to further develop the technology in hope that this system could be applied to help reducing the attrition rate of beginning CS students.

"Responding to broad user interest and with the NSF's encouragement, the two started Turing's Craft in early 1999 to commercialize the WebToTeach technology and thereby make it broadly available. In June 2000, Turing's Craft received a technology commercialization grant from the Empire State Development fund in recognition of the outstanding potential of the technology. In the spring of 2002, Turing's Craft released the commercial version of WebToTeach, CodeLab." [5]

3.2 Practice-IT

The Practice-It tool lets you type code into a browser form and submit it, runs tests on it, and shows the results in the browser. Students are requested to fill in the body of a method to solve a problem. This online tool allows you to practice problems from the textbook Building Java Programs (2nd edition) and from the University of Washington's introductory Java programming courses authored by Marty Stepp and Stuart Reges. [6]

3.2.1 History

The Practice-IT website was written by Marty Stepp and Jessica Miller, instructors of Computer Science & Engineering at the University of Washington. Practice-IT was inspired by similar tools such as CodingBat, TuringsCraft CodeLab, Lrn2Java, and JavaBall. The majority of the problems on the website are copyrighted by Pearson Education. The University of Washington takes the pedagogical approach of objects late in their CS1 course so the emphasis is on procedural programming in Java. Classes are not introduced until late in the course. The problems used in the CS2 course at Washington focuses on usage and implementation of data structures, OOP, and algorithms. [6]

3.2.2 Homework Assignments

There are currently 456 total problems implemented and growing as instructors

are allowed to create their own and add it to the web site.. Practice-It has a variety of CS1 problems shown in Figure 3 [6].

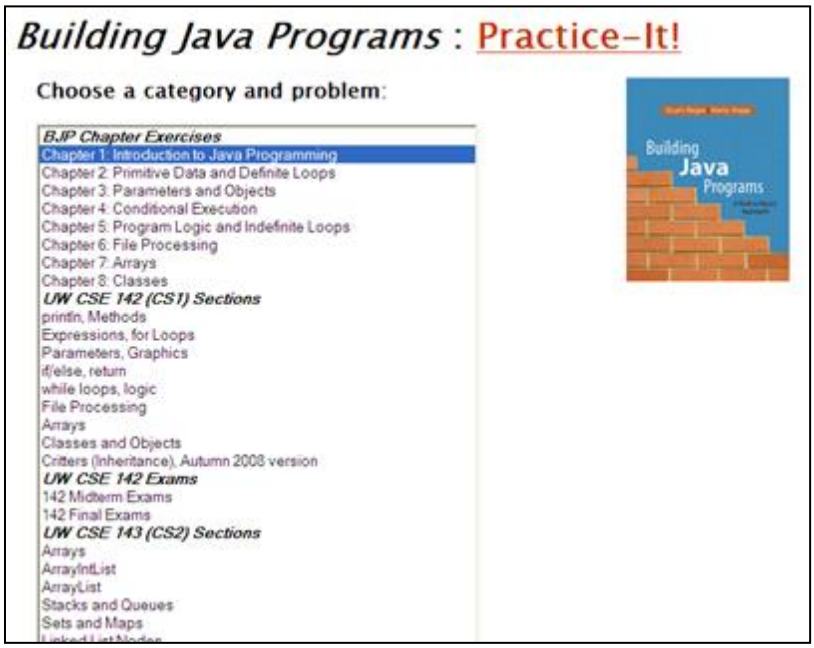

Figure 3: Practice-IT GUI

which includes: strings, arrays, recursion, OOP, selection structurs (if, case), repetition structures (for, while, do-while), random numbers, parameter passing, and file processing. A typical problem is shown in Figure 4.

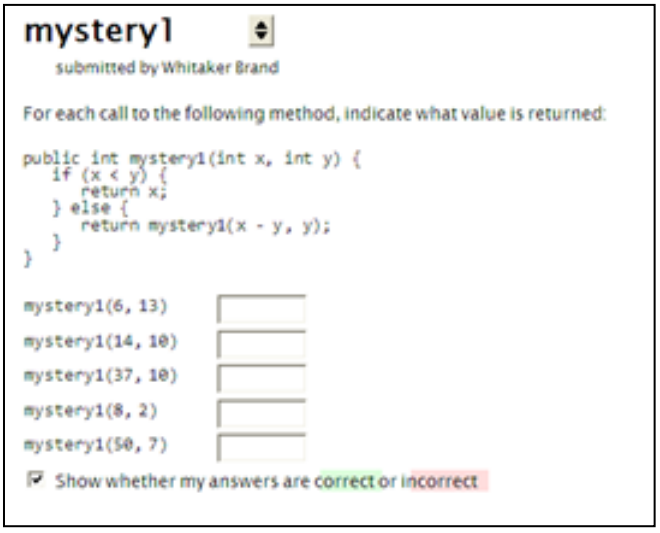

Figure 4: Practice-IT Exercise

3.2.3 New Features

Some recent features added to Practice-IT are a timed quiz feature which allows students to take a practice exam and see if they can finish all of the problems in a given amount of time. The system now remembers your solution code for each problem you solve and re-displays it when you return to a given problem. The problem set has been reorganized for better structure. Now, if you want to use Practice-IT, students must create an account to log in order to use the tool. [6]

3.3 CodingBat

CodingBat is a free site of live coding problems to build coding skill in Java, (also in Python) created by Nick Parlante who is a computer science lecturer at Stanford. The coding problems give immediate feedback, so it's an opportunity to practice and solidify understanding of the concepts. The problems could be used as homework, or for self-study practice, or in a lab, or as live lecture examples. The problems, have low overhead: short problem statements (like an exam) and immediate feedback in the browser. [7]

3.3.1 Background

The idea for CodingBat came from Nick's experience teaching introductory computer science courses at Stanford along with his observations of upper-level computer science students use unit-tests in the advanced courses. After watching Owen Astrachan, an instructor at Duke, demonstrate a unit-testing "thing" [7] that he uses with his Duke students, the idea of an online tool came to fruition. The first online tool was named JavaBat and with the addition of Python exercises, the online tool was renamed to be CodingBat. The GUI is shown in Figure 5.

Figure 5: CodingBat GUI

3.3.2 Pedagogical Approach

In order to succeed in writing programs to solve problems, students have to build skill in both the "large" and the "small." The concept of "large" is defined to be the larger issues in computer science where students need experience in algorithms and data structures to solve larger problems which students achieve from the entire computer science curriculum. In order to get to that skill level, students also need skill "in the 'small' which is defined as 10 or 20 line methods built of loops, logic, strings, lists etc. to solve each piece of the larger problem. " [7] If students do not have success achieving skills in the "small" (loops, logic), there is no chance for them to achieve success acquiring skills in the "large".

Students need practice writing methods and this tool gives them plenty of opportunities. This repeated process helps reinforce concepts delivered in lecture which allows student to understand the larger programming assignments that are assigned. A typical exercise is shown in Figure 6 and the solution is shown in Figure 7.

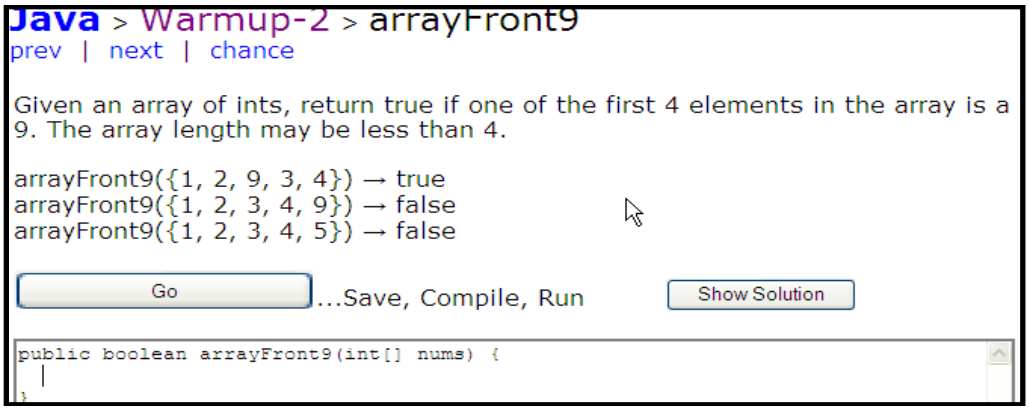

Figure 6: CodingBat Exercise

Solution:

```
public boolean arrayFront9(int[] nums) {
  // First figure the end for the loop
  int end = nums.length;
  if (end > 4) end = 4;

  for (int i=0; i<end; i++) {
    if (nums[i] == 9) return true;
  }

  return false;
}
```

Figure 7: CodingBat Solution

3.3.3 New Features

The CodingBat website just introduced the following new features to aid instructors: [7]

1. Authoring
2. Report Page

Instructors are now allowed to create their own exercises and include those in the assignments for their students. The report page now features a filter control so you can select between java or python student problems, and be able to choose between problems that originated with CodeLab and those that the instructor had custom written.

4 Comparisons of Online Tools

As these online tools mature, the major differences between the tools are decreasing. However, there are several significant features that still separate these online tools. The online tools are compared in three major categories: classroom logistics, grading, and homework exercises.

4.1 Classroom Logistics

It is very important the any tool used in the course does not increase the time demands on the instructor to administer the tool. Figure 8 on the next page shows the various categories used for classroom logistics.

4.1.1 Cost

The cost for any tool can be a major issue for financially challenged students. The fact that Practice-It and CodingBat are free sets both of them apart from CodeLab. The cost of Codelab is $30 per student which the student pays by credit card or pay pall on the website. If the instructor can purchase for the entire course, then access can be granted for $20 per student.

4.1.2 Student Accounts and Instructor Account

CodeLab offers the easiest path to create student accounts and assign them to your course. An instructor requests a CodeLab class account via email and receives an URL to post for their students that helps them register them for the course. For Practice-IT and CodingBat, the instructor and students create accounts off the web site. They both have features that allow the students to share their results with their instructor.

Online Tools	Classroom Logistics		
	Cost	Account Creation	Alignment with Textbook
CodeLab	$20-$30	Student/ WebSite	Website
Practice-IT	Free	Student/	Instructor
CodingBat	Free	Student	Instructor

Figure 8: Classroom Logistics

4.1.3 Alignment with the Textbook

CodeLab allows the instructor to specify in the email the course textbook and the exercises in CodeLab will be assigned to the various chapters in your textbook. The instructor can request which problem numbers appear for each chapter. Not all textbooks are currently supported.

For Practice-IT and CodingBat, the instructor would have to assign the different homework based on the various chapters which increases the initial setup for the course. In addition, whenever any new exercise is submitted to the website, the instructor must assign that exercise to the chapter that covers that concept. The online tool Practice-IT is tightly aligned with the accompanying textbook by the same authors who developed the online tool.

4.2 Course Grading

It is very important the any tool used in the course eases the burden of grading on the instructor. Figure 9 shows the various categories used for course grading. In

CodeLab, the instructor requests the roster which keeps track of the problems solved by the students and keeps running totals. The instructor receives a comma delimited spreadsheet which can be uploaded to the Course Management System being used for the course. The instructor cannot assign point values on the web site as each exercise is assumed to be worth the same value.

CodingBat is the only online tool which allows instructor to set higher point values for certain exercises including zero points for exercises that are used for demonstration purposes. CodingBat does not have any convenient way to update the instructor's course management system. Coding Bat allows the student to set a preference so the instructor can view on the webpage the problems solved and the solution that the student used to solve the problem. Practice-IT has no reporting mechanism to the instructor to use for a grade for the course which is significant. Students are more likely to use the online tool if they receive credit for their work.

Online Tools	Grading			
	Grading Feature (P/F)	Assign Point Values	Reporting Mechanism	CMS Alignment
CodeLab	Yes	No	Yes	Partial
Practice-IT	Yes	No	No	No
CodingBat	Yes	Yes	Yes	No

Figure 9: Grading

4.3 Homework Exercises

The primary purpose of using online tools is allowing students to get additional practice on learning key concepts and solving problems. Figure 10 lists the categories rated for Homework Exercises.

Online Tools	Exercises						
	Amount Of Exercises	Concepts	CS1 Problems	CS2 Problems (Data Structures)	Solutions	Help - Hints - Clarifications	Ability to Add Exercises
CodeLab	300+	Yes	Yes	Limited	All	Yes	Yes
Practice-IT	350+	No	Yes	Yes	Partial	Partial	Yes
CodingBat	Smaller	No	Yes	Yes	Partial	Partial	Yes

Figure 10: Exercises

4.3.1 Amount and Coverage of Exercises

Both CodeLab and Practice-IT have a larger set of exercises which makes sense as both of these online tools were created before CodingBat. All of the online tools allow instructors to add new exercises to the web sites.

4.3.1.1 CS1 Exercises

Both Practice-IT and CodingBat exercises involve implementing the required method. CodeLab has exercises for smaller concepts like declaring variables and assignment statements as well as larger concepts of creating classes.

4.3.1.2 CS2 Exercises

However, CodeLab only has exercises which can be found in various CS1 courses so the tool does not have any beginning Data Structure exercises which are usually covered in CS2 available. Both Practice-IT and CodingBat have beginning CS2 data structure exercises available.

4.3.2 Solutions and Help

CodeLab has solutions for all the exercises. Both CodingBat and Practice-IT only have solutions for many of the exercises but not all of them. CodeLab has help available for every exercise while CodingBat and Practice-IT have hints for the majority of the exercises.

5.0 Conclusion

The emphasis for finding online tools has been to use them to augment teaching CS1 courses both traditional and distance education. CodeLab is the only commercial product which is both an advantage and a disadvantage. An instructor is very comfortable that the website will be maintained and available for students to practice. It would be nice to find a free online tool to use so students would not have to find additional fees for the course. CodeLab is the only tool that offers elementary concepts to the development of classes. For this reason, CodeLab has been my choice of online tools and will continue to be my choice until I can find a free online tool that can replace the advantages of CodeLab.

References

[1] UW System, Two-Year Campuses
http://www.wisconsin.edu/campuses/twoyear.htm

[2] Zweben, S., 2003-2004 Taulbee Survey, *Computing Research News, Vol. 17/No.3*, May 2005. http://www.cra.org/CRN/articles/may05/taulbee.html

[3] UW – Colleges,
http://www.uwc.edu/

[4] Turing's Craft – The Exercises,
http://www.turingscraft.com/exers.php

[5] http://turningscraft.com

[6] http://webster.cs.washington.edu:8080/practiceit/index.jsp

[7] http://codingbat.com/