USING TECHNOLOGIES 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. 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.

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 teaching the CS1 course. This core Computer Science course is usually a programming class with student drop-out rates higher than the discipline would like to see. Now what happens if this CS1 course must be offered via distance education?

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 (Figure 1). 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 13,500 students.

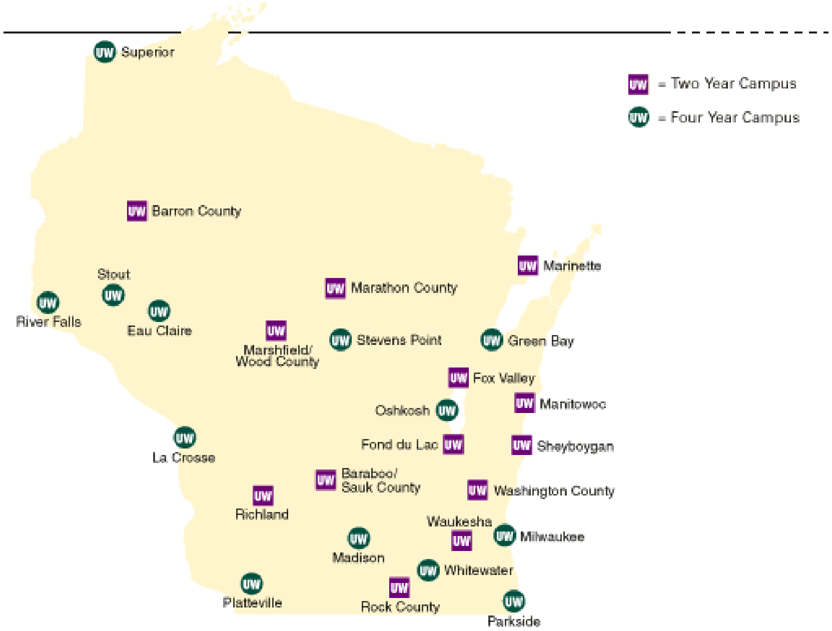


Figure 1: Campuses of the UW-System

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. Already, the CS2 and CS3 core courses have to be offered via distance education for all of the 13 UW-College campuses due to low enrollment. The smallest 5-7 campuses (Figure 2) based on enrollment do not have sufficient enrollment in CS courses to offer the CS1 core class so their campuses were not offering the course and, thus denying those students the chance to take their first CS course. The loss of potential CS1 students at five campuses also meant there were less potential students to fill the CS2 course which could put that in danger of being cancelled due to low enrollment.

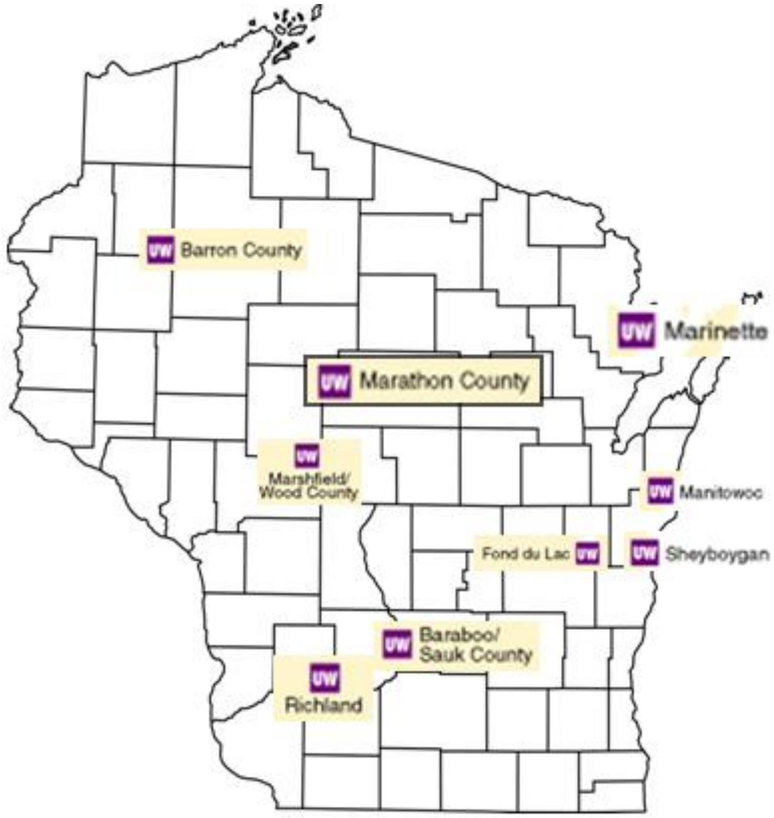


Figure 2: Distance Education campuses for CS1 course

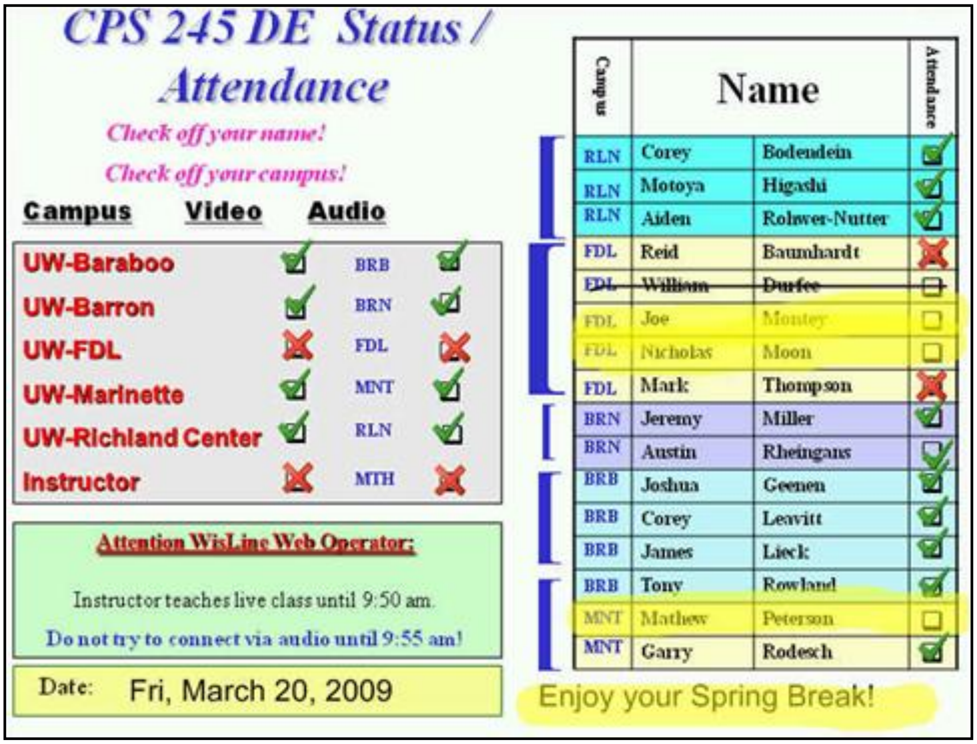
1.3 Time on Task

Approximately 70% of all students admitted to the UW-Colleges graduate in the bottom half of their high school graduating class. It is easy to understand that these students are classified as at-risk students to finish a college degree. The majority of these students do not have the necessary math requirements to enroll in the CS0 course, let alone the CS1 course. They are forced to take remedial math courses prior to actually taking their first CS course. It is easy to see 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.

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.

1.4 CS1 Course Overview

The course meets for 50 minutes, three times a week, for 15 weeks. There is a 2 hour mandatory lab so each campus has a lab instructor. Some campuses have problems finding a qualified lab instructor. A typical lecture starts with the campus status slide (Figure 3) that students update when they sign-in so the instructor can see if there are any video or audio issues along with checking attendance.



CPS 245 DE Status / Attendance

Check off your name!

Check off your campus!

Campus	Video		Audio
UW-Baraboo	✓	BRB	✓
UW-Barron	✓	BRN	✓
UW-FDL	☒	FDL	☒
UW-Marinette	✓	MNT	✓
UW-Richland Center	✓	RLN	✓
Instructor	☒	MTH	☒

Attention WisLine Web Operator:

Instructor teaches live class until 9:50 am.

Do not try to connect via audio until 9:55 am!

Date: Fri, March 20, 2009

Campus	Name		Attendance
RLN	Corey	Bodendein	✓
RLN	Motoya	Higashi	✓
RLN	Aiden	Rohwer-Nutter	✓
FDL	Reid	Baumhardt	☒
~~FDL~~	~~William~~	~~Durfee~~	□
FDL	Joe	Montey	□
FDL	Nicholas	Moon	□
FDL	Mark	Thompson	☒
BRN	Jeremy	Miller	✓
BRN	Austin	Rheingans	✓
BRB	Joshua	Geenen	✓
BRB	Corey	Leavitt	✓
BRB	James	Lieck	✓
BRB	Tony	Rowland	✓
MNT	Mathew	Peterson	□
MNT	Garry	Rodesch	✓

Enjoy your Spring Break!

Figure 3: CS1 Status/Attendance Slide

Labs specifications and a Lab grading rubric are created and agreed upon by all the lab instructors. The campus lab instructor grades the labs for those students on their campus.

Students only receive one grade as the lab grade is factored into the overall grade. Besides 15 labs, there are three programming projects that all students must implement and turn in to the main instructor to grade. This is done to ensure that all students are being graded consistently for these programming projects. A programming project would typically incorporate concepts that were covered for previous 4-5 labs.

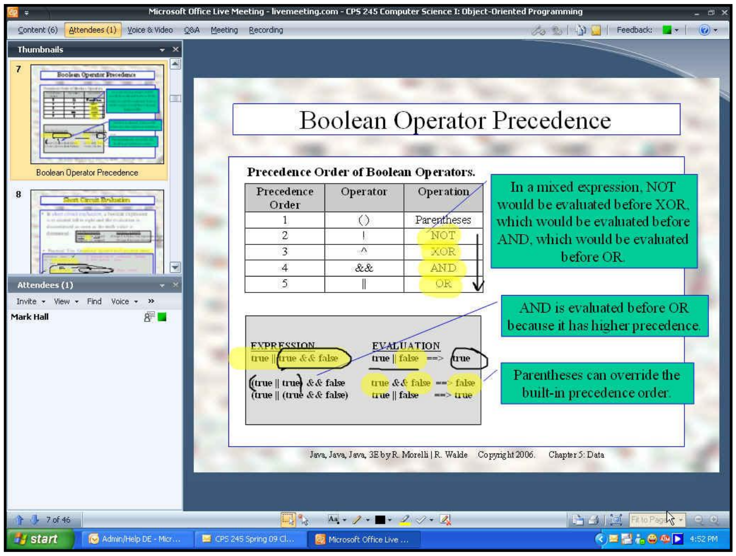
2 Technologies Used For DE Course

The following technologies are used to enhance the Distance Education experience:

- WisLine Web – Microsoft Live Meeting
- Desire-To-Learn (D2L) Course Management System [2]
- CodeLab [3]
- Instant Messaging

2.1 WisLine Web – Microsoft Live Meeting

WisLine Web is powered by Microsoft Office® Live Meeting – a Web-based application [3]. Students can see the contents of my PC screen projected on a local large-screen display (Figure 4). Audio is over a bridged telephone hook-up so we can hear each other so the class can be interactive.



Boolean Operator Precedence

Precedence Order of Boolean Operators.

Precedence Order	Operator	Operation
1	( )	Parentheses
2	!	NOT
3	^	XOR
4	&&	AND
5	||	OR

In a mixed expression, NOT would be evaluated before XOR, which would be evaluated before AND, which would be evaluated before OR.

```
EXPRESSION                      EVALUATION
true || true && false           true || false   ==>  true
(true || true) && false         true && false   ==>  false
(true || (true && false)        true || false   ==>  true
```

AND is evaluated before OR because it has higher precedence.

Parentheses can override the built-in precedence order.

Java, Java, Java, 3E by R. Morelli | R. Walde Copyright 2006. Chapter 5: Data

Figure 4: Layout of WisLine Web Meeting

2.1.1 Physical Requirements

To participate in a WisLine Web meeting, all that is needed is: [3]

- PC running Windows 98 or newer; Mac OSX 10.1 or newer using Safari
- Internet connection (56k or faster)
- Web browser (IE 5 or newer)
- Telephone with mute feature

2.1.2 Teaching Tools

WisLine Web options for content delivery include:

- PowerPoint® slides
- Annotation tools
- Live software demonstrations (TextPad)
- Polling (per campus, not per student)
- Whiteboards
- Web tours, and
- Graphic snapshots

Presenters have full control of when and how content is delivered. There are also a variety of tools for presenters to control and receive interaction. Slide content can be made available to participants to save as a pdf file and print from their local computers. All lectures are recorded as a streamed media file. Archives are available on the web for two weeks unless requested for a longer period of time. Archived lectures can be used by the students to view for the following reasons:

- Campus closure due to weather
- Student absence due to sickness/emergency
- Students with learning disabilities to view multiple times

The instructor's audio is always live while students must press the microphone button in order to communicate with the instructor. The instructor cannot see the students nor can the students see the instructor.

2.1.3 Lecture Preparation

UW-Colleges contract out for the Live Meeting Server which is not hosted locally, but in another state. The instructor must upload his slides to the server prior to using them in the class lecture. On the day of the class, I sign in to the server several hours prior to class, locate and update the status/attendance slide, and then locate and find the slide to start for the current concept being discussed. Then I go back and display the status/attendance slide so that will be displayed when class starts. If any Java code is to be discussed in class, then I load those files into TextPad so it is ready to be shared during the class.

Five to ten minutes prior to class, the distance education campuses DE representative or the students sign-in to the Live Meeting Server, and update the status/attendance slide for their campus so when class starts, I have an understanding of any technical issues for a campus, and have an attendance record for the students. Students must show up at the appropriate DE classroom on their campus in order to attend the lecture. The audio is automatically connected for each site regardless if anyone is in attendance.

During the lecture, students see my slides as I discuss them, and I can annotate the slides on the fly to force discussion questions. My audio is live while students must press the microphone button to speak. I can also share an application, usually TextPad to display any Java code.

2.2 Desire-To-Learn (D2L) Course Management System

The UW-System has chosen the Desire-To-Learn (D2L) Course Management System to be used on all UW campuses [4].

2.2.1 What is a Course Management System (CMS)

(CMS) are specific types of Knowledge Management (KM) systems often used in academic settings. Examples of software applications for creating a CMS include Blackboard, Web CT and Desire-To-Learn.

2.2.2 Advantages of CMS

The following are common advantages that can be found in most course management systems:

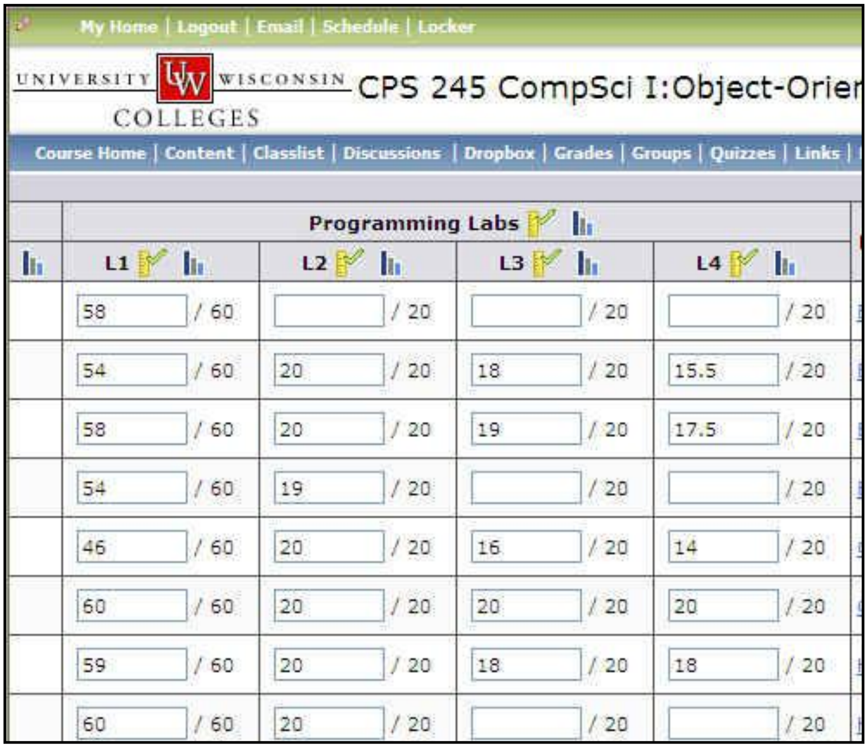
- Reasonably user friendly.
- Support student collaboration and dialogue.
- Support links to documents, images and other web sites.
- Provide a student roster with email capabilities.
- Facilitate real-time "classroom" discussions.
- Facilitate on-line testing of students.

2.2.3 Common Features

Some common features that can be found in these course management systems are: Grade Book; Content area for Assignments and Information, Chat, Discussion Forum, Quizzes, and Surveys. The features that are used in this course that I want to emphasize are the grade book and the discussion forums.

2.2.3.1 Grades

This feature is a no-brainer when it comes to distance education. The instructor can immediately post grades for exams, labs, and homework assignments. Students can immediately know their overall grade status. After posting the results on D2L, I send out an email to let the students know that grades have been posted.


My Home | Logout | Email | Schedule | Locker
UNIVERSITY UW WISCONSIN
COLLEGES
CPS 245 CompSci I:Object-Orier
Course Home | Content | Classlist | Discussions | Dropbox | Grades | Groups | Quizzes | Links |

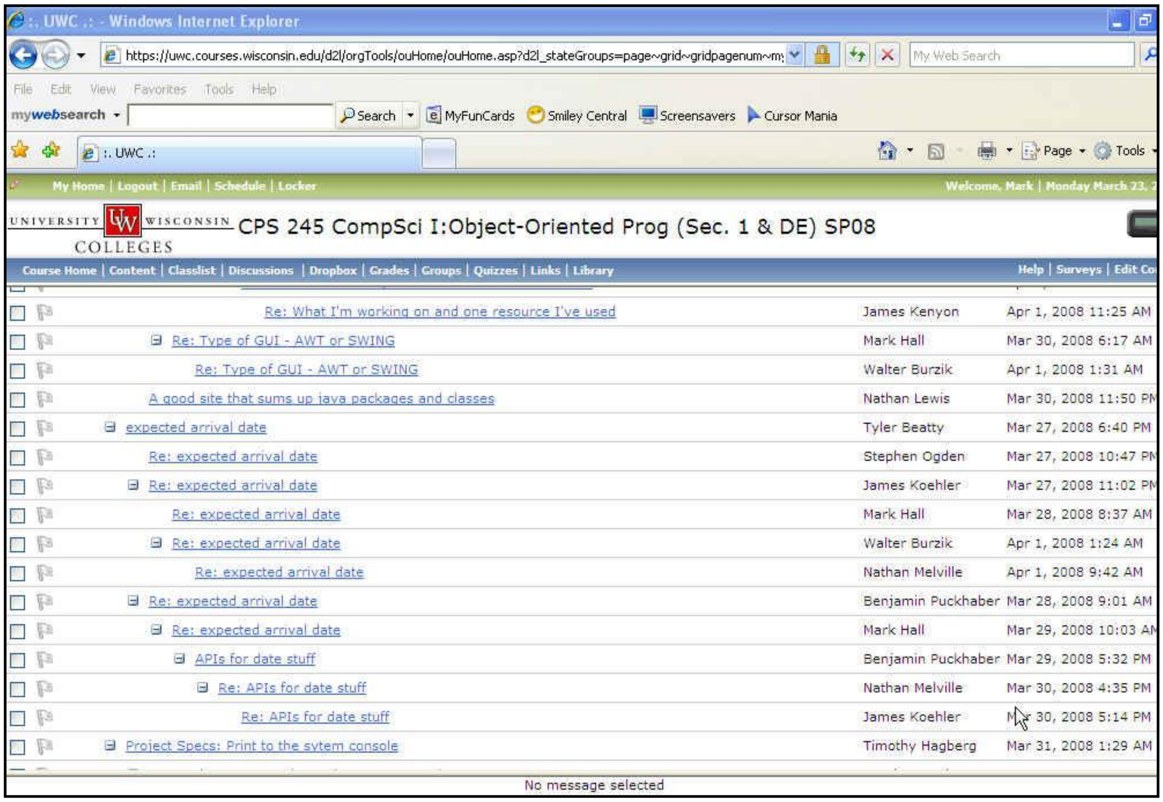
	Programming Labs			
	L1	L2	L3	L4
	58 / 60	/ 20	/ 20	/ 20
	54 / 60	20 / 20	18 / 20	15.5 / 20
	58 / 60	20 / 20	19 / 20	17.5 / 20
	54 / 60	19 / 20	/ 20	/ 20
	46 / 60	20 / 20	16 / 20	14 / 20
	60 / 60	20 / 20	20 / 20	20 / 20
	59 / 60	20 / 20	18 / 20	18 / 20
	60 / 60	20 / 20	/ 20	/ 20

Figure 5: D2L Grade Book

2.2.3.3 Discussion Forums

The CS1 course uses the discussion forum (Figure 6) so that all students can establish a virtual class where students can help each other for labs and/or programming projects. This allows students to not feel like they are alone on the campus if they have problems. The rule is that students can discuss design concepts at any time if their post is in English, not java. Complete code assignments cannot be posted, and code snippets can be posted if approved by the instructor prior to posting.

Each programming project has a discussion forum participation grade as part of the overall grade. Participation includes posting questions and replying to postings whether the posting contains the answer, or asking for more clarification. As the instructor, I try to allow time for students to reply postings before I post a response. Lab instructors are encouraged to read and reply to postings also.


:. UWC .: - Windows Internet Explorer
https://uwc.courses.wisconsin.edu/d2l/orgTools/ouHome/ouHome.asp?d2l_stateGroups=page~grid~gridpagenum~m;
My Web Search
File Edit View Favorites Tools Help
mywebsearch
Search
MyFunCards
Smiley Central
Screensavers
Cursor Mania
:. UWC .:
Page
Tools
My Home | Logout | Email | Schedule | Locker
Welcome, Mark | Monday March 23, 2
UNIVERSITY UW WISCONSIN COLLEGES
Course Home | Content | Classlist | Discussions | Dropbox | Grades | Groups | Quizzes | Links | Library
Help | Surveys | Edit Co

CPS 245 CompSci I:Object-Oriented Prog (Sec. 1 & DE) SP08

□	Re: What I'm working on and one resource I've used	James Kenyon	Apr 1, 2008 11:25 AM
□	⊟ Re: Type of GUI - AWT or SWING	Mark Hall	Mar 30, 2008 6:17 AM
□	Re: Type of GUI - AWT or SWING	Walter Burzik	Apr 1, 2008 1:31 AM
□	A good site that sums up java packages and classes	Nathan Lewis	Mar 30, 2008 11:50 PM
□	⊟ expected arrival date	Tyler Beatty	Mar 27, 2008 6:40 PM
□	Re: expected arrival date	Stephen Ogden	Mar 27, 2008 10:47 PM
□	⊟ Re: expected arrival date	James Koehler	Mar 27, 2008 11:02 PM
□	Re: expected arrival date	Mark Hall	Mar 28, 2008 8:37 AM
□	⊟ Re: expected arrival date	Walter Burzik	Apr 1, 2008 1:24 AM
□	Re: expected arrival date	Nathan Melville	Apr 1, 2008 9:42 AM
□	⊟ Re: expected arrival date	Benjamin Puckhaber	Mar 28, 2008 9:01 AM
□	⊟ Re: expected arrival date	Mark Hall	Mar 29, 2008 10:03 AM
□	⊟ APIs for date stuff	Benjamin Puckhaber	Mar 29, 2008 5:32 PM
□	⊟ Re: APIs for date stuff	Nathan Melville	Mar 30, 2008 4:35 PM
□	Re: APIs for date stuff	James Koehler	Mar 30, 2008 5:14 PM
□	⊟ Project Specs: Print to the sytem console	Timothy Hagberg	Mar 31, 2008 1:29 AM

No message selected

Figure 6: D2L Discussion Forum

2.3 CodeLab

CodeLab [5] is the commercial version of an academic NSF project[6], WebToTeach [7, 8]. 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 exercise is presented to the student and the student types in their solution to the exercise. 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.”[9]

2.3.1 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.” [9] 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.” [9]

2.3.2 Benefits of using CodeLab

Both the students and the instructor can benefit from using CodeLab for the CS1 course.

2.3.2.1 Benefits to the Student

There are inherent communication problems when teaching distance education courses. CodeLab allows students the ability to work on homework assignments and helps reduce student frustration as students receive immediate feedback on their answers. Students cannot walk down the hall to talk to the professor since they are on another campus. In addition, sometimes there are only one or two students on the campus so they do not have the opportunities to discuss any problems with their peers.

In CodeLab, since every exercise comes with feedback, hints and other explanatory material, the brick wall syndrome should be reduced, and sometimes even eliminated. Since CodeLab gives feedback within seconds of the solution submission, students can quickly understand if they know this concept. CodeLab has multiple assignments covering the main concepts. However, students still have the ability to discuss the assignment with the professor during class or outside of class via a phone call, email, or instant messaging.

2.3.2.2 Benefits to the Instructor

CodeLab aids the students in learning the syntax, semantics and the basic usage of the programming language. The instructor can now spend class time can now be spent on problem solving strategies, design and analysis, abstraction, algorithms, and style. This emphasis can aid in the development of the critical thinking skills of the students.

Since there are no Teaching Assistants at the UW-Colleges, it is very easy to feel overwhelmed with the tasks of grading homework and programming assignments. CodeLab automatically checks student work for correctness and tracks student performance by maintaining their submission records which eliminates a major task for the instructor. In order for CodeLab to be effective, a grade for CodeLab must be part of the overall syllabus, otherwise students will not spend much time using CodeLab. In this CS1 course, students must complete 90% of all CodeLab assignments to receive the maximum homework points which constitute 10% of their grade. CodeLab allows me to view all exercises for all students, whether completed or not. The exercises are aligned with the chapters of the textbook so I set deadlines that those exercises must be completed prior to the exam in order to receive credit. This deadline is necessary to force students to keep current.

2.3.3 CodeLab Exercises

CodeLab exercises are short, focused on a particular topic and automatically evaluated. The exercises range from "one-liners" like variable declarations, arithmetic and boolean expressions, assignment statements to more complex problems involving loops, functions, methods or even small class definitions. Each concept has multiple exercises so that the concept is reinforced to aid in learning. This forces students to spend more time on task.

2.3.3.1 Student Feedback

The following are some student comments from the CS1 course when asked to evaluate CodeLab:

- “It is very helpful to be able to do the simpler examples, and then gradually move to more difficult examples, it helps with basic concepts.”
- “The examples are very consistent in format.”
- “For beginning programmers it provides a good base knowledge, and if something is not known the user can find it in the book relatively easily. “
- “In my personal opinion, CodeLab is a very useful online resource.”
- “A major advantage that is found in CodeLab is its feature of notifying the user of a correct or incorrect submission.”
- “Chapter 7: Equal (20598) - After trying to use the equals sign with respect to string several times in my programs and having it not compile this was very useful.”

- "I liked these problems because they made a relation with one another. Each one build on the previous one and it gave a better understanding of how the interaction works in Java."

2.3.3.2 Examples

Every semester, I ask students to give me three CodeLab exercises that they thought were useful, and three that they thought were not useful. Several exercises are mentioned every semester.

2.3.3.2.1 CodeLab Exercise #20649

"max is a method that accepts two ***int*** arguments and returns the value of the larger one. Four ***int*** variables, ***population1***, ***population2***, ***population3***, and ***population4*** have already been declared and initialized. Write an expression (not a statement!) whose value is the largest of ***population1***, ***population2***, ***population3***, and ***population4*** by calling ***max***."

A student commented that he "liked this one because it showed that you can call a method with the same method in the parameters. And I don't recall going over this in lecture, so I learned from this exercise."

2.3.3.2.2 CodeLab Exercise #20622

"Write an ***if/else*** statement that compares the double variable ***pH*** with 7.0 and makes the following assignments to the ***int*** variables ***neutral***, ***base***, and ***acid***:

0,0,1 if pH is less than 7
0,1,0 if pH is greater than 7
1,0,0 if pH is equal to 7"

A student commented that "although the directions weren't every clear I did like this one. This one was a bit more challenging and I had to think about it where as some of the exercises were very easy. I learned more about selection statements and how they can be used."

2.4 Instant Messaging

In order to be available to students when they have questions, my PC at my office and my laptop at home are always signed in to the following IM applications:

- Microsoft Messenger
- Yahoo! Messenger
- AOL Messenger (AIM)

Each one of these IM applications are free text-based messaging tools that also support sending text messages and files. My IM applications are set to send SMS messages to my cell phone if I am not signed in to that application. Students are forced to pick an IM application and make contact with me as part of a homework assignment the first week. This guarantees that students know that they can quickly make contact with me for any quick question.

3 Other Possible Technologies

Some other technologies are being tested to see if they could be used to support this CS1 course. They are:

- Breeze to replace WisLine Web
- Practice-It to replace CodeLab
- Student Response Systems (or clickers) to replace Polling (Live Meeting)

3.1 Adobe Connect (Breeze Meeting)

Adobe Connect is a web conferencing application that offers a way to bridge the physical and virtual classroom space by using real-time audio and video, chat tools, and application and file sharing. Adobe Acrobat Connect Pro 7 allows educators to teach in real time, or deliver narrated presentations and courses that students can view on-demand. Either way, educators can capture and keep the attention of learners with engaging video, audio, and interactive multimedia course material that requires no additional downloads.

Further, Adobe eLearning solutions provide dynamic virtual classroom management tools that increase instructor efficiency [7]:

- Breakout rooms allow expanded curriculum and more specific focus.
- Customizable layouts and persistent content reduce preparation time.
- Instructors can reuse and archive virtual classroom settings with editable online and offline recordings.
- For class assignments, and for out-of-class feedback on those assignments, Acrobat software delivers an electronic PDF workflow, complete with easy-to-use commenting tools.

Some educational uses of Adobe Connect are:

- Conduct meetings and/or office hours (on or off campus) via a chat session that could include audio, images, Web pages and PowerPoint files
- Present material to students who may not have been able to attend class; sessions can be recorded for later viewing
- "Bring in" outside experts or guest lecturers into the classroom
- Small group collaboration on projects

Breeze allows voice over IP (VIP) so that students could actually attend class from their home without having to drive to their campus.

3.2 CodeLab Possible Replacement (Practice-It)

Marty Stepp at the University of Washington has developed a web-based Java practice problem tool called Practice-It which is meant to accompany his new Java CS1 textbook, Building Java Programs that he wrote with Stuart Reges. 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. It is still in a *"beta"* state and missing some features that would be necessary before I would switch, like login and the ability to keep track of problems attempted and solved for individual students [8].

There are currently 357 total problems implemented. Practice-It has a variety of CS1 problems which includes:

- Strings
- Arrays
- Recursion
- Object-oriented programming
- Selection Structures (if/else mystery problems)
- Repetition Structures (for and while)
- random numbers
- parameter-passing
- file processing

3.3 Student Response Systems (SRS or Clickers)

The use of "clickers" has been documented as another technology tool that can be a powerful interactive learning system. The UW-Colleges attempted to implement clickers for the WisLine Web mode of teaching distance education. The server that holds the content is not collocated with the UW-Colleges so that the clickers cannot be used with the current mode of PowerPoint slides. However, the instructor could share a local PowerPoint application that could be used to ask questions to the remote campuses. If the instructor also tries to display his instructor slides locally, the network bogs down as the slide must be transferred to the Live Meeting server and then transferred to each campus.

The software from Turning Technologies, LLC transforms the standard lecture PowerPoint® presentation into a very interactive learning system that engages students in active participation as instructors can display questions, surveys, opinion polls and games that force students to respond [9]. This could replace the Polling feature of WisLine Web which can only capture a campus response versus individual students. These clickers can track attendance and assess student learning retention which instantly informs the professor if the students really understand the concept that was just presented.

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] WisLine Web FAQ
http://www.uwex.edu/ics/faqpub/faq_a.cfm?faq=487
[4] Desire-To-Learn (D2L)
http://www.desire2learn.com/
[5] Turing's Craft
http://turningscraft.com
[6] Turing's Craft – The Exercises
http://www.turingscraft.com/exers.php
[7] Breeze
http://www.adobe.com/education/pdf/elearning/ED_elearningSolutionBrief.pdf
[8] Practice-It
http://webster.cs.washington.edu:8080/practiceit/
[9] Turning Technologies, LLC
http://www.turningtechnologies.com/highereducationinteractivelearning.cfm