References


High failure rate in computer science are a big problem. One of the biggest problems is syntax and syntax related errors. A solution is to introduce students to problem-solving skills before applying those skills in a programming environment.


A new system is implemented based on WebToTeach. The new environment is used to take test on the web. The most useful part of this short paper is the comments on security.


This paper talks about the design decisions and performance of WebToTeach. It talks about the services used and the process to set up exercised within the environment.


Looking at the move to object oriented programming.


All aspects of Computer Science education are moving to the web. The biggest benefit is moving from static to dynamic teaching methods. It also provides a non-localized environment. Problems with the static teaching methods are discussed. Current implementations show the active movement to the web paradigm, introducing interactive, dynamic, animated exercises.

A whole course is outlined including a schedule, reading assignments, assignments, and deadlines. The assignments use CRC-Cards and BlueJ. An evaluation is performed looking at quiz and exam scores, followed by student feedback.


Web-service simulations yield significantly better learning performance than environments. Students have difficulties in an e-learning environments and belief in problem solving influences their performance.


Students grade 7 and younger learn better in a graphical environment while senior programmers learn better in a textual environment. Students between these grade levels find graphical environments too easy and textual environments too difficult. Thus, the BrickLayer environment is recommended to ”fill the gap” so these students can have a more efficient learning environment.


Concludes that TDD needs to be taught in the curriculum due to the results of several studies. The question becomes, how and when?


Teaching programming is not enough. Educational institutions must also teach testing strategies to give students automated evaluation of their correctness and validity.


This paper summarizing 5 novice programming environments based on anecdotal, analytical, and empirical techniques. A rubric is created for evaluating these systems.


An online course is developed for high school students. This course can help students get into 3 universities in Finland. The course provide some requirements for an environment. The environment should support group activities, encourage students to work collaboratively, interact with tutors online, and graphically make learning computer science more fun than just normal text based programming systems.


This paper addressed difficulties that students encounter and how BlueJ attempts to solve those problems. Students use the environment and perform an evaluation on BlueJ. Students rated BlueJ a mean of 4.3 on a scale of 1-7, where 7 is low.


Students using test-first strategies write more test and are more productive.


Industry that uses TDD passed 16%-50% more external test cases. Provides educators with empirical evidence to put TDD in the curriculum.

Using TDD provides better code. This study shows metrics on productivity, code length, coverage, and complexity.


A method for teaching TDD is proposed. Shows that test-first gives students slightly better comprehension at no extra cost.


Shows statistics and metrics on TDD in industry. In general, programmers using test-first strategies wrote smaller units that were less complex and highly tested.


Students using test-first strategies have been shown to be more productive and average a whole letter grade higher than students using test-last strategies.


A HUGE taxonomy of CS environments ranging from super simple systems to very complex. Also looks at systems for all grade levels, as well as old and new systems. This paper provides a great compilation of the environments with corresponding features and categories.


TDD has become one of the most used practices in XP. This study runs two experiments in master courses and discuss the pros and cons.
This paper analyzes teaching object-oriented programming in the CS curriculum. This paper will be used to make the analogy that TDD is like OO, and that we need to teach it.

BlueJ’s website which includes downloads for the BlueJ environment.

This paper describes the features and pedagogy of BlueJ. This paper is useful because it gives specific assignment sequences; starting with steps, and then the overall picture.

This paper discusses the history of OO design and the paradigm shift to OO. It then describes the features of BlueJ. An interaction and experimentation study is done resulting in positive feedback.

A short overview of the benefits and results of using online tutors. The results show a significant change in performance.

Identifies and addresses problems with languages and tools in undergraduate classes. An environment is created to address broad language support, integrated tools, consistent user interface, and efficient and portable implementations. The environment is geared for all students of all levels in an undergraduate degree.

A multi-national, multi-institutional study of assessment of programming skills of first-year CS students.

An in depth study is performed to determine the skills of first-year CS students. This paper provides useful methodologies, assessment techniques, and guidelines for environments. It also provides great charts of their results.


Gives an overview of IDE, optimizations, and user interface of CodeLab. Gives good examples of exercises used in the environment.


Watson attempts to help Computer Science majors and non Computer Science majors to learn computing. Watson provides 12-15 exercises, each with a specific purpose. Exercises geared for majors will concentrate more on programming concepts. Exercises geared for non majors will focus on computer application concepts. Watson identifies requirements for entry level learning environments with a few flaws.


The first paper published on the ELP that gives an overview of the system, including design decisions. An evaluation of the ELP is done with a survey to 12 students.


Discusses the current status of the Environment for Learning to Program (ELP) used by Queensland University of Technology. They cover the reasons behind the implementation, compare it to other tools, and explain the results from an experiment where students used the ELP.

Describes the features and framework of the ELP. The paper compares the ELP to different environments. Final, it does a case study on 39 students during the 5th week or a course using the ELP.


Website for Codelab. Gives information on the environment and free demos to instructors.


This paper goes into deep detail of all the BlueJ features. It also gives a break down of student usages between features and an evaluation on speed and performance.


12 factors influence a students success when learning computer science. Comfort level is the best predictor of success in learning computer science. Comfort level does not “cause” success but trying to make students comfortable will improve a students performance.