

115 East 57th Street
10th Floor
New York, New York 10022
212.515.6200 phone
212.515.3624 fax

CognitiveArts

February 9, 2001

Professor Moshe Vardi
Department of Computer Science
PO Box 1892
Rice University
Houston, TX 77251-1892

Dear Professor Vardi:

I am writing to offer you a solution to a standard problem in computer science departments across the country. Who will teach the introductory programming courses?

I have spent the last thirty years of my career, first at Stanford, then Yale, and most recently at Northwestern University, exploring and developing learning environments that are reflective of how people naturally learn. We have had tremendous success developing powerful simulation based software for leading corporations, including GE, IBM, and Merrill Lynch. Our efforts in the corporate community have provided us with the experience to prove our approach of goal-based learning and take those lessons to the students I care about most, yours.

Columbia Interactive Arts & Sciences, is a unique online collaboration between Columbia University and Cognitive Arts Corporation. We are building a variety of on line, learn by doing courses, working in close cooperation with Columbia. We have just made these courses available to the public and can now make them available to your students as well. The computer courses we have available are C++, JAVA, and HTML. These courses were built as part of Columbia's Computer Technology Applications (CTA) Program in the Division

Chicago
Boston
New York

of Continuing Education and thus have a more practical bent. The Columbia CTA Courses are designed to immerse students in authentic simulations that reflect what an IT job is really like.

These courses cover the same basic computer science concepts and skills that you would expect in any serious college-level programming course. But we believe they do much more than that. Unlike other courses, both distance-based and classroom-based, our courses present everything in the context of the software development life cycle. We don't just talk about how software is developed, we put the students on the job. From Day One the student plays the role of a member of a corporate programming team. Task assignments come from a team manager, and detailed specifications from a lead programmer, with frequent references to client needs, concerns, and reactions. Students get to see where real programming problems come from, how API's are used to support team programming, and how code review is used to monitor and improve code quality, particularly maintainability.

Then we add something students rarely get in college and fear to get on the job: in-depth code review and critiquing. Every piece of code a student submits is subjected to an detailed analysis. Code is examined for inefficiencies, subtle bugs, maintainability problems unnecessary complexities, poor design and so on. Problematic code is returned to the student with detailed annotations to be reworked and resubmitted.

This "do-review-redo" process means our students do more than learn enough to "make it work." They learn how to "do it right," how to write readable code, how to choose the right constructs for each task, how to use functions and classes to organize code into maintainable modular units, and so on.

Each course consists of approximately 14 major task assignments. These assignments begin with simple graphics and numeric calculations and work up to an interactive calculation and graphical design tool. Along the way, the students learn core concepts in the following areas.

Basic Concepts and Skills: Our introduction to programming courses cover the elements common to what the ACM used to call CS 1, i.e., variables, functions, loops, arrays, structures, and input/output, both console and file.

Object-oriented Analysis and Design: Both courses are object-oriented from the start. Students are designing and implementing classes by the midpoint of each course. In C++ they learn how to separate API and implementation in header files and code files. In Java, they learn about public versus private methods.

Iterative Development: Students learn the concepts and skills through a series of tasks that lead to the construction of a large application, that combines graphics, numerical calculations, and file I/O. Rather than independent, isolated, homework assignments, students see how large projects can be constructed as a series of approximations that are tested, review and extended.

Modern Programming Platforms: Students on PC's use Microsoft's Visual C++ integrated development environment for C++ programming and Borland's JBuilder Foundation for Java programming. Students on Macintoshes use Metrowerks' CodeWarrior Pro for C++ and Java.

Portable Standard Code: Students are taught to code to language standards, avoiding platform-specific extensions, using the current C++ Standard Library and the Java 1.2 libraries.

Cross-platform Simplified Graphics: In both courses, students develop graphical interfaces from the start, as well as the more traditional text-based interfaces. Small simple graphical libraries hide complicated platform-specific details from beginning programmers so that they can focus on the key concepts of screen coordinates, shape drawing, and event handling.

Throughout these courses the student is actively engaged in building a suite of graphical software for a client. They advance at their own pace, but they are not left to struggle on their own. The student has the guidance and support of a Tutor who has experience as a computer programmer working in the IT industry. And they are able to access online resources designed to answer virtually any question that might arise.

At the completion of each of the courses, your students will receive a letter of completion along with an official Columbia University transcript. More IT courses to complement the introductory courseware will be available throughout 2001, along with courses in English as a Second Language, Psychology, Physics, and Economics.

If you are interested in learning more about the courses soon to be available, or have any questions, please feel free to contact me directly via e-mail at schank@cognitivearts.com. I look forward to hearing from you. Many thanks.

Sincerely,

A handwritten signature in black ink, appearing to be 'R. Schank', written over the printed name.

Roger Schank
Chairman and Chief Technology Officer
Cognitive Arts Corporation
Professor Emeritus of Computer Science
Northwestern University