I wrote the following three essays as part of the application for the 2002 Radio Shack National Teacher Awards.

John Hanna
Published to the WWW April 29, 2002

John Hanna

<u>Question One</u>: *Describe the ideal use of technology (broadly defined by you as a teacher) to:*

- *access learning,*
- *improve the quality of learning in the classroom, and*
- *support innovation*

"Technology" is the means by which society provides itself with things. Inventions such as the printing press brought literacy to the common folk. More recently, the invention and widespread use of pens and pencils made the school slates obsolete. Today, calculators and computers pervade our sense of "technology".

In an academic setting, technology should support curriculum, not supplant it. As we shift from a product-based to a service- and knowledge-based economy, facility with computers and the Internet, and proficiency with devices like graphing calculators and other handheld computers becomes critically important for all members of the academic community. But we must not lose sight of the broader goals of curriculum: to give students the knowledge and skills that they need to be productive members of a rapidly advancing society; to give them an understanding of our histories, cultures and human experience; to give them the insight and wisdom to make sound, informed decisions; and to prepare them to live cooperatively yet competitively in an ever-changing, increasingly diverse community.

Schools must integrate the skills and concepts of "Computer Applications" into the entire curriculum. *All* teachers must be, at the very least, 'competent' in the use of the computer in the classroom, especially the 'basics' of word processing, spreadsheet, and presentation software, and the skills concomitant with Internet-based research. *All* teachers must have convenient access to Internet-enabled computers and have access to student information in the district database. *All* students should regularly present computer-based projects illustrating curricular topics, and we should expect students to use technology often, effectively, and appropriately.

Computer Science is a misunderstood field. School administrators, guidance counselors, and even most teachers do not appreciate the curriculum because it is clouded in jargon. Teaching and learning computer programming is a valuable experience that gives students practical problem-solving skills, deeper insight into the technology that they use, and challenging concepts that apply to all sorts of career avenues, from business to science and engineering. The administration must support the Computer Science curriculum, expanding, and refining it to reach as many students, especially females, as possible. Women are badly under-represented in the Computer Science field and increasing their numbers in our Computer Science program is one of my goals.

Handheld technology has come a long way since 1988, when the first graphing calculators became available. Soon, wireless "personal digital assistants" (like Palm Pilots) will replace these calculators, making their application in the curriculum more widespread. This will give both students and teachers easy access to information at the "tap of a stylus". Our Professional Development programs must now address this technology proactively and prepare teachers for the next round in "technology infusion" before the students surpass us.

John Hanna
<u>Question Two</u>: *Describe your favorite lesson plan (1 day minimum – 1 week maximum) that actively involves your students.*

As a Mathematics teacher *and* a Computer Science teacher, I look for challenging, interesting programming projects that are multi-tiered and appropriate for the students' *mathematical* competence. One such project involves the *divisibility* properties of the Natural numbers.

The **Divisibility** project bases all its components on any programming language's capability of handling either integer division or the MOD operator. Divisibility is tested in different ways in different languages. Some languages do not have a MOD operation (C++ and Java use "%"), so the expression "*x/n* = int*(x/n)*" was and is used to test for divisibility of *x* by *n*. The project consists of eight components, each of which builds on prior knowledge and skills:

1. Write a program that allows the user to enter two numbers and displays whether or not the first is divisible by the second
2. Write a program that uses a loop to display all the divisors of an entered number
3. Write a program that displays the number of divisors of an entered number
4. A ***PRIME*** number has exactly two divisors: 1 and itself. Write a program that displays whether or not an entered number is ***PRIME***
5. Write a program that uses nested loops to display **a list of prime numbers** up to an entered number
6. Write a program that **adds up the proper divisors** of an entered number and displays this total. ***Proper divisors*** do not include the number itself
7. A ***PERFECT*** number equals **the sum of** its proper divisors. Write a program that uses nested loops to display the ***PERFECT*** numbers up to an entered number
8. Two numbers are ***AMICABLE*** ("friendly") numbers if the proper divisors of one add up to the value of the other. Write a program that finds and displays a pair of ***AMICABLE*** numbers, both of which are less than 1000

Each component of the project builds on the one before it. As a student progresses through the project, she adds to the program using the common thread of 'divisibility.' In this way, a structured approach to building a large programming project is introduced gradually and effectively.

This project is assigned in the late first or early second marking period in a computer science course. It is easily adapted for any programming language, and I have used it for years in BASIC, Pascal, C++, and now, Java programming courses. Because of its mathematical slant, it also addresses certain local, state, and national mathematics competency standards. It challenges students to build on prior learning, reinforcing the fact that learning does not take place in isolation – it is a hierarchal process.

Depending on the level of the course, the above project will take from ½ week to 2 weeks to complete, assuming that I have about 3 to 4 hours of contact time each week.

John Hanna

Question Three: *If you could focus on one project or area of investgation for the next school year:*

- *what would you study*
- *how would you implement your new knowledge in the classroom, and*
- *what specific audience segment(s) would be impacted positively?*

The "Infinity Project" is an introductory course in engineering focusing on electrical engineering using computers and Digital Signal Processors. The project's home page is www.infinity-project.org. I first saw the idea presented at an exhibitor's booth at a T^3 conference two years ago. Last summer, I attended a one-week teacher training institute at Southern Methodist University in Dallas and was immediately hooked. As I wrote in my evaluation, "This is the most exciting project I've encountered since my introduction to the graphing calculators."

This year, I have six students working on the Infinity Project materials and computer projects on an 'independent study' basis. They all have copies of the book and I have purchased six Infinity Project hardware kits consisting of the Digital Signal Processor hardware and the Visual Application Builder software.

My focus for next year is to get an Introduction to Engineering course implemented at Teaneck High School. We have the facilities and the student talent to offer such a program. The program readily and clearly addresses local, state, and national standards in mathematics, science, and technology. The program gives students exposure to the *Engineering Design Method* and shows them the relationship among the strategies of the *Programming Design Method* and the *Scientific Method* from those disciplines.

The curriculum uses the textbook Multimedia and Information Engineering, by Orsak, Athale, Douglas, Munson, Treichler, Wood, and Yoder (Prentice Hall, 2002). Lab activities include real-time audio and video signal processing experiments using (a) built-in sound and video capabilities of an ordinary personal computer, (b) Digital Signal Processor hardware provided by Texas Instruments, Inc. (www.ti.com), and (c) unique computer 'programming' software called "Visual Application Builder" from Hyperception, Inc. (www.hyperception.com).

It is clear to me that this kind of high-tech curriculum is needed to challenge students and to get them thinking about a career in engineering during their high school experience. But even if their career goal is uncertain, the kinds of problem-solving strategies that the curriculum employs will aid students in the pursuits of many varied goals. Most Fortune 500 CEOs are actually engineering graduates! We need to do all we can to maintain the supply of talented, resourceful and educated managers. As Geoffrey Orsak, project director, stated: "Either get on this [engineering] bandwagon or become irrelevant." I intend to remain relevant.