CS 6704: Developing Computational Thinking in Middle School Curricula

"Do you think you can teach someone to think like a computer scientist?"

INTRODUCTION

This class consists of a few small teams (of students in CS 4784, the HCI Capstone class), led by graduate students (from this class), working together with teachers to identify and prototype classroom activities that give middle school students the opportunity to engage in computational thinking.

"Computational thinking" is the approach to problem-solving used by computer scientists. A computational thinker has a mindset that promotes:

- Methods for analyzing and logically organizing data
- Locating issues in modeling, abstraction, and simulation
- Formulating problems such that computers may assist
- Identifying, testing, and implementing possible solutions systematically
- Automating solutions via algorithmic thinking
- Generalizing and applying this process to other problems

The faculty on this project believe that there are opportunities to locate computational thinking in different subject areas of middle school education. And those opportunities will result in new, additional ways for middle school students to master core subjects and become more rigorous and critical thinkers. We seek a few good students who are willing to explore this.

COURSE DESCRIPTION

This class is a special graduate seminar. This is a project-based class in software engineering, focusing on problem finding and requirements development. Students will work directly with middle school teachers to identify, specify, and prototype an activity that teaches content-area material that also uses computational thinking. The project will have multiple faculty members who are working on the project advising and directing the students.

There will be a single 3-hour class meeting each week, possibly in the early evening. (Probably Tuesday 6 to 9 PM.)

Students enrolled in the project can expect to take three to five trips to Henrico County to observe classrooms and interact with teachers. Travel will be organized and reimbursed by the project.

Because the project is teaching "participatory design" which is an alternative software design methodology, this is an advanced topic in software engineering.

LEARNING OBJECTIVES

Having successfully completed this course, the student will be able to:

- Design a solution to a significant open-ended problem in the area of computer science education.

- Develop strategies for innovation involving experts in the learning sciences and computer science along with educational practitioners and students.
- Evaluate or assess a proposed solution to a problem in this domain.
- Function effectively in teams, identifying the issues of problem finding, problem solving, values, team-work, ideation, presentation, and critique when working across disciplinary boundaries.
- Coordinate meaningful design representations across various modes and media.
- Document and present (using written, oral and visual means) the design process and the results from a proposed solution to a problem in this area.

PROJECT BACKGROUND

Henrico County Public Schools

This is the second semester of a 2-year project working with Henrico County Public Schools (outside of Richmond). The district has a large commitment to computers in classrooms and a program to advance its curriculum in all subjects to take advantage of the technology. More importantly, they have identified what they call "21rst Century" skills that make use of computing technology and critical thinking.

Computational thinking

There are many different ideas as what constitutes "computational thinking". Part of the objective of this project is come up with experiences that promote some aspect of computational thinking. Here are a few ideas about what constitutes computational thinking:

Jeanette Wing:
http://www.imageofcomputing.com/pdf/Computational_Thinking.pdf

- Abstraction
  - C.T. is operating in terms of multiple layers of abstraction simultaneously
  - C.T. is defining the relationships the between layers
- Automation
  - C.T. is thinking in terms of mechanizing the abstraction layers and their relationships (Mechanization is possible due to precise and exacting notations and models)
  - There is some "machine" below (human or computer, virtual or physical)

Wikipedia:

- Analyzing and logically organizing data
- Data modeling, data abstractions, and simulations
- Formulating problems such that computers may assist
- Identifying, testing, and implementing possible solutions
- Automating solutions via algorithmic thinking
- Generalizing and applying this process to other problems

Microsoft and CSTA:
http://education.sdsc.edu/resources/CompThinking.pdf
The essence of computational thinking is thinking about data and ideas, and using and combining these resources to solve problems.

Our grant

At the time of writing this class proposal, we have a grant from the CE21 program in the CISE directorate of the NSF called “Planning Grant: Integrating Computational Thinking into the Middle School Curriculum”.

WHO SHOULD TAKE THIS CLASS

Admission to the class will be by interview with one of the instructors. Programming skills are, of course, important. However, knowledge of problem finding and requirements development is also valuable. Therefore, we are looking for students who have taken Introduction to HCI, GUI programming, and/or Software Engineering.

Students with existing connections in the Richmond area may find it easier to accommodate the travel requirements of the class.

FACULTY

Deborah Tatar and Steve Harrison will jointly teach the class. However, other members of the CE21 project will actively engage with the students as appropriate. They are:

- **Chris Corallo** (Ed.D. Virginia Tech) Executive Director of Organizational Development, Quality, and Innovation for Henrico County Schools in Richmond, Virginia.
- **Dennis Kafura** is Professor of Computer Science at VT. He will advise on approaches to thinking about computational thinking.
- **Manuel Perez-Quinonez** is Professor of Computer Science at VT. He is interested in how our approach to computational thinking can broaden participation.
- **Cliff Shaffer** Professor of Computer Science at Virginia Tech, has an extensive history with the design and development of technology used to promote CT at the university level.

SYLLABUS

Structurally, the class is part seminar, part lecture, and mostly hands-on project development. Grading will be based on seminar preparation and participation (20%), a paper in “CHI” conference format describing the project (25%), and the design and implementation of the project (55%).

Content of Course

1. Orientation and methodology	10

2. Technical skill development	5
3. Project planning & management	5
3. Principles of the learning sciences	10
4. Problem finding in domain area	15
5. Creative design solutions (ideation)	20
5. Implementation	30
6. Evaluation	15
____ Percent of course	100