Introducing Computational Thinking in Education Courses

Aman Yadav
Dept of Educational Studies
Purdue University
West Lafayette, Indiana, USA
amanyadav@purdue.edu

Ninger Zhou
Dept of Educational Studies
Purdue University
West Lafayette, Indiana, USA
zhoun@purdue.edu

Chris Mayfield
Dept of Computer Science
Purdue University
West Lafayette, Indiana, USA
cmayfiel@cs.purdue.edu

Susanne Hambrusch
Dept of Computer Science
Purdue University
West Lafayette, Indiana, USA
seh@purdue.edu

John T. Korb
Dept of Computer Science
Purdue University
West Lafayette, Indiana, USA
jtk@purdue.edu

ABSTRACT

As computational thinking becomes a fundamental skill for the 21st century, K-12 teachers should be exposed to computing principles. This paper describes the implementation and evaluation of a computational thinking module in a required course for elementary and secondary education majors. We summarize the results from open-ended and multiple-choice questionnaires given both before and after the module to assess the students' attitudes toward and understanding of computational thinking. The results suggest that given relevant information about computational thinking, education students' attitudes toward computer science becomes more favorable and they will be more likely to integrate computing principles in their future teaching.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Curriculum

General Terms

Experimentation, Measurement

Keywords

Computational Thinking, K-12 Education, Non-Majors

1. COMPUTATIONAL THINKING IN K-12

Wing suggested that "computational thinking" (CT) is a fundamental skill of analytical thinking for everyone, not just for computer scientists [10]. She described computational thinking as "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science." Wing also pointed out the untapped potential of computational thinking for K-12 education by stating, "To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability."

A report on computational thinking by the National Research Council (NRC) advanced a similar idea, that CT is a cognitive skill which the "average person is expected to possess" [5]. Similarly, Bundy suggested that computational thinking concepts have been used in other disciplines via problem solving processes, and that the ability to think computationally is essential to every discipline [4]. The pervasiveness of computational thinking concepts dictates the importance of exposing students to such notions early in their school years and helping them to become conscious about when and how to apply this essential skill.

The NRC report also highlighted "(1) that students can learn thinking strategies such as computational thinking as they study a discipline, (2) that teachers and curricula can model these strategies for students, and (3) that appropriate guidance can enable students to learn to use these strategies independently." Teacher education is one discipline where computational thinking will have significant impact. As we prepare future educators to present their subject areas using ideas from computational thinking, K-12 students will have greater exposure to computing in general.

In this paper, we describe the implementation of a CT module in two sections of a core education course required for all elementary and secondary education majors. We present a pre- and post-assessment of the education students' understanding and attitude of computational thinking, which measured the influence the CT module had on them. In particular, only 20% of the students on the pre-survey described computing as, "the process of solving problems" as compared to 70% on the post-survey. On the pre-survey, 30% of the students agreed that computing relates to any or all fields, and the percentage increased to 62% on the post-survey. Overall, the CT module helped the students understand that (1) they can teach computing concepts in K-12 classrooms without the use of computers, and (2) CT concepts can be incorporated across all disciplines.

Computational thinking in education has the potential to significantly advance the problem-solving skills of K-12 students. However, literature on implementing computational thinking in a K-12 setting is still relatively sparse. There

SIGCSE'11, March 9–12, 2011, Dallas, Texas, USA.

have been a number of workshops on integrating computational thinking at the high school level [1, 6, 7]. But there is little research that has systematically and comprehensively examined the influence of computational thinking on pre-service teachers (i.e., education students).

Exploratory investigations have demonstrated how exposure to computational thinking enhances the way students approach problems. For example, Lewandowski et al. illustrated the idea of "commonsense programming" for students without programming experience [8]. Students were asked to propose solutions to avoid selling theater tickets for the same seat twice at multiple box offices. The results showed that 69% of the solutions correctly identified a race condition, which indicated that the students were indeed equipped with a natural but undeveloped understanding for solving problems computationally.

Several researchers have made the effort to introduce CT to in-service computer science high school teachers during workshop sessions that promote the awareness of integrating CS across all subject areas [3]. The high school teacher participants were reported to have extended their understanding of the scope of CS and considered CS as more than just programming. More importantly, a comparison of the pre- and post-survey indicated that the realization of the importance of "developing computational thinking skills for all aspects of life" emerged as a result of the workshop.

An important step for successfully integrating computational thinking into the K-12 curriculum is to prepare future teachers to teach it. Section 2 briefly describes our efforts to showcase a variety of CT concepts to pre-service teachers. In Section 3, we present a summary of our pre- and post-survey, which demonstrates a positive change of attitude among the students. We conclude the paper with an overall discussion in Section 4.

2. OVERVIEW OF THE CT MODULE

We developed a one-week computational thinking module for the course "Learning and Motivation," which is required for all elementary and secondary majors. The course introduces future K-12 teachers to basic concepts of classroom management, learning styles, student motivation, and assessment. The main content of the course includes theories of learning and motivation, the role of formal and informal assessment in fostering learning and motivation, and ways to adapt instruction both to individual students and differences in social, cultural, and contextual factors.

Since computational thinking naturally includes problem solving and understanding human behavior, it fits well with the topics already covered in class, such as probabilistic reasoning, algorithms and heuristics, and hypothesis testing. We replaced the originally presented unit on problem solving and critical thinking with new lectures on computational thinking. The purpose of the CT module was not only to expose students to ideas in computing, but to show how these ideas can be used in their future teaching careers as well.

Faculty and students from educational studies and computer science jointly developed the lecture material. Special emphasis was placed on highlighting the core concepts of computational thinking, while presenting material the students could relate to and easily apply in a K-12 classroom. The module was presented around the middle of the semester, and students had not been exposed to CT material in earlier lectures. Before the CT module was introduced, students had primarily studied educational theories on how people learn.

The CT lectures provided students with an overview of computational thinking and engaged them in activities that showcased CT principles. Students worked in pairs and each pair was given a "clicker" to provide responses. See the appendix for an outline of the lectures and clicker questions. The original slides are available on our website[1].

The first lecture introduced students to the definition of computational thinking and five basic CT concepts: problem identification and decomposition, abstraction, logical thinking, algorithms, and debugging. The concepts were introduced through examples, activities, and clicker questions. For example, debugging was discussed by asking students to troubleshoot the scenario of a lamp not working when they get home from school, but was working in the morning. Students used clickers to respond to a sequence of questions to locate the problem in order to make the lamp work again.

The second lecture focused on the role of computational thinking in day-to-day life, and emphasized the importance and application of CT in K-12 education. We discussed how problem-solving, abstraction, and critical thinking can be introduced in a classroom setting. The lecture highlighted how computational thinking is a useful tool for dealing with ill-defined problems, where there might not be a clear-cut solution and information needed to solve the problem may be missing. The lecture also presented information on how to teach algorithms through kinesthetic activities and gave an example of recursion. Several members of our project acted out the Towers of Hanoi (for $n = 5$). Finally, we provided examples of computational thinking in core content areas, such as science and humanities.

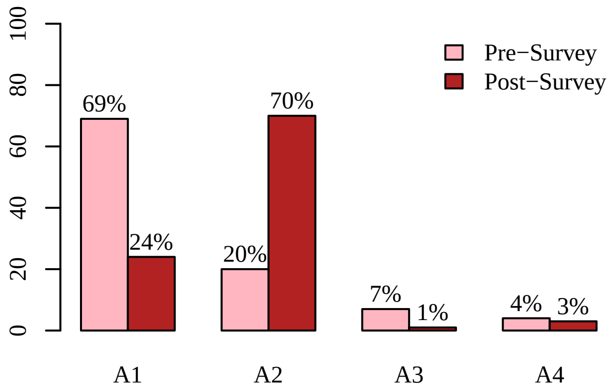
3. ATTITUDE SURVEY RESULTS

We assessed the students' attitudes toward computing and their understanding of computational thinking through a pre- and post-survey surrounding the CT module. All 155 students from the two sections of the course were emailed a link to the survey. 100 students — 78 female and 22 male — completed both the pre-survey the week before the CT module and the post-survey during the week after the CT module. As a motivation to participate, extra credit was given to students who completed both surveys (the pre- and post-surveys were anonymous, but linked together via answers to security questions).

The surveys consisted of sixteen multiple-choice questions (on a Likert scale from strongly agree to strongly disagree) and four open-ended questions — see the appendix for the complete post-survey. An external evaluator was responsible for collecting and validating the data. We analyzed the responses for patterns and significant differences between pre- and post-surveys. The most interesting results came from the participants' open-ended responses, which we present in this section.

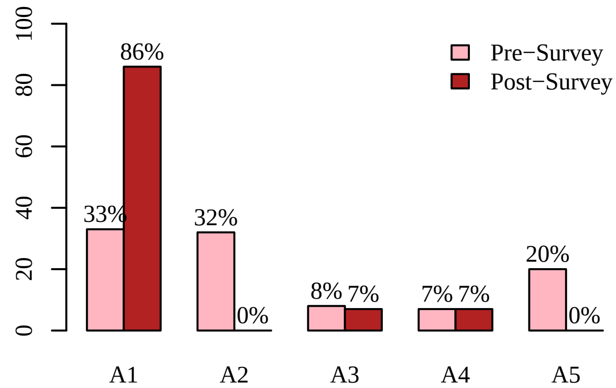
Just over half of the students who responded (55) were preparing to teach at the elementary level; the remainder (45) at the secondary level. Approximately 80% of the elementary education participants were female (44 vs. 11), while at the secondary level, over 75% were female (34 vs. 11). About 65% of the students enrolled responded to the survey.

[1]http://cs4edu.cs.purdue.edu/comp_think



A1: To use computers and/or technology to solve a problem and make tasks easier
A2: The process of solving problems (use of computer or technology not mentioned)
A3: The study of computers
A4: Other

Figure 1: Participants' view of computing



A1: The process of solving problems
A2: To use computers and/or technology to solve a problem or make tasks easier
A3: The study of computers; solving problems like a computer
A4: Other
A5: Not sure

Figure 2: Participants' view of CT

3.1 Participants' View of Computing

Survey participants were asked to describe their view of computing and its purpose. The basic trends of the participants' responses differed greatly from the pre- to post-survey, as shown in Figure 1. (Not all percentages add up to 100% due to blank responses.)

On the pre-survey, a majority (69%) of the participants' responses contained themes that viewed computing as *solving problems or making tasks easier through the use of computers and/or technology*. For example, one student stated, "Computing is the use of computers or some other form of technology to solve a problem. Its purpose is to help solve problems that might be more difficult to solve without some form of technology." Only 20% of participants' responded viewing computing and its purpose as *the process of solving problems (use of computer or technology not mentioned)*. The following comment highlights computing as a process of solving problems without the use of computers or technology: "Computing is the science of solving a problem using some pre-set method that has been established. Its purpose is to assist us in solving everyday problems we might be faced with." Finally, 7% of the students reported computing as the study of computers. For example, "Being able to use and apply computer skills to daily life. These skills can run from typing to using different applications and software in many different areas."

In the post-survey, a majority of the participants' responses (70%) reflected the basic trend that viewed computing as *the ability/knowledge/process used to solve problems and make tasks easier (use of computer or technology not mentioned)*. For example, a student highlighted this view of computing stating, "Computing is where you logically think something through. Its purpose is so that students can explore to find an answer to a problem." Another student reported, "I believe that computing does not necessarily involve working with computers but working to solve any type of problem. Its purpose is to calculate equations and problems to formulate a correct result." On the other hand, 24% viewed computing as *solving problems or making tasks easier through the use of computers and/or technology*. This view is highlighted by the following comments: "Computing is using technology or computer software to aid in solving problems. Its purpose is to make difficult tasks easier and simpler." Only one participant reported computing as the study of computers.

By de-emphasizing technology and encouraging algorithmic thinking, the lectures helped the education students understand that computational thinking doesn't always require the use of computers to solve problems.

3.2 Participants' View of CT

In both the pre- and post-survey, participants were asked to define computational thinking from their point of view. Figure 2 summarizes the basic trends in the participants' responses between the pre- and post-surveys. Although answers varied, we found basic trends among them. One third (33%) of participants' responses in the pre-survey reflected trends that viewed computational thinking as "the process of problem-solving," which increased to 86% on the post-survey. In addition, responses used more specific terminology and examples (e.g., pre-survey response: *process of solving problems*, post-survey response: *process of solving problems using algorithms, etc.*).

On the pre-survey, 20% of participants responded that they "didn't know" what their view on computational thinking was, and on the post-survey, none of the participants responded that they "didn't know." Additionally, on the pre-survey, where almost one third (33%) of participants' responses reflected trends regarding computational thinking as "the use of computers or technology to solve a problem or make tasks easier," none of the participants' responses on the post survey reflected the idea that computers and technology absolutely must be involved in the definition of computational thinking.

In short, many more students identified the relationship between problem solving and computational thinking after receiving the CT module.

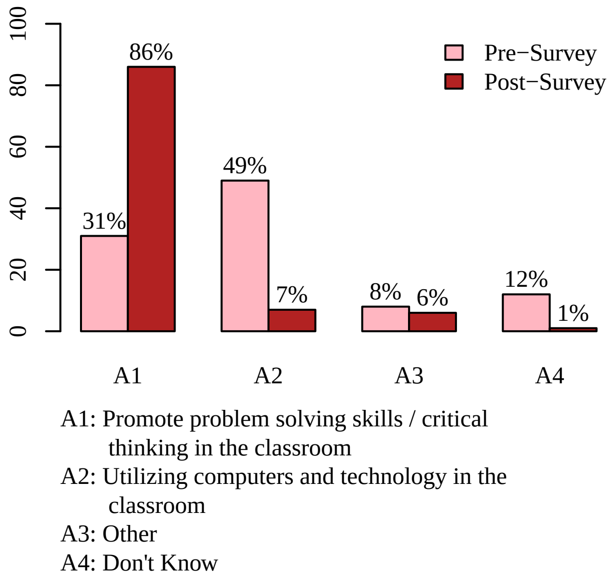


Figure 3: Integrating CT into the classroom

3.3 Integrating CT into the Classroom

Survey participants were also asked how computational thinking can be integrated into the K-12 classroom. Figure 3 summarizes their responses. On the pre-survey, almost half of the participants' responses reflected the view that in order to integrate computational thinking into the classroom, computers and technology were needed. In contrast, only 7% of participants' responses reflected that trend on the post-survey.

During the one week module, many students realized the benefits of understanding CT principles and being able to apply them more systematically as a problem solving technique. This change in attitude is also reflected in the question on whether CT promotes problem solving and critical thinking skills. The responses switched from 31% agreeing to 86% of the participants agreeing. Also, it seems every student had some concept about computational thinking after the module, as reflected by the "don't know" category.

In summary, we saw almost a three-fold increase in the number of education majors who now hold the opinion that computational thinking could be used to enhance problem solving activities in a K-12 classroom.

3.4 Relationship to Other Fields

We also asked the students about the relationship of computational thinking to other disciplines. While 30% indicated that "Computational thinking relates to any or all fields" in the pre-survey, that number rose to 62% in the post-survey. In addition, over 95% of the survey participants either agreed or strongly agreed with the following statements: "Computational thinking *can* be integrated into classroom education in other fields," "Computational thinking *should* be integrated into classroom education for other disciplines," and "Having background knowledge and understanding of computer science is valuable in and of itself."

As future work, we plan to embed computational thinking modules in content area courses and teaching methods courses, including hands-on training in how to implement kinesthetic activities like *Computer Science Unplugged* [2].

4. DISCUSSION

Results from the two surveys suggest that the CT module was effective overall in increasing the students' awareness of computational thinking. Specifically, the post-survey responses were more sophisticated and showcased students' understanding that computational thinking was more than using computers and technology. Students also had a better grasp of how computational thinking can be integrated into their future teaching by promoting problem solving and critical thinking skills (i.e., not by merely using computers).

These findings have important implications for incorporating computational thinking in education as well as other subject areas. Given that computational thinking is becoming a fundamental skill for the 21st century, it is important to introduce it in disciplines outside of computer science and at the K-12 level. Specifically, computational thinking concepts must appear as early as the primary grades, and then continue through the secondary grades and beyond [9]. One way to do this is to incorporate computational thinking modules into core education courses to expose future teachers to this idea. Results from the current work suggest that such an approach has the potential to change future teachers' understanding of computational thinking and how it can be integrated in their classrooms.

In summary, we have shown that given relevant information in computing, education students' attitude toward computing becomes more favorable. They also see applications of computing principles in their careers more readily. We plan to repeat the CT module in "Learning and Motivation" and are also developing an online version of the module for "Introduction to Educational Technology," another course required of all education majors. We hope that our approach to CT modules will attract education students into computing courses that emphasize computer science principles, as well as traditional programming courses.

5. ACKNOWLEDGEMENTS

This work was supported by the NSF CPATH program under grant CNS-0938999 and through a gift from State Farm. We especially thank Courtney Brown for her help with the administration of the two surveys. We also appreciate the CS4EDU team and the anonymous reviewers for their thoughtful feedback on the paper.

6. REFERENCES

[1] V. Allan, V. Barr, D. Brylow, and S. Hambrusch. Computational thinking in high school courses. In *SIGCSE*, 2010.

[2] T. Bell, I. H. Witten, and M. Fellows. Computer science unplugged. http://csunplugged.org/, December 2006.

[3] L. Blum and T. J. Cortina. CS4HS: An outreach program for high school CS teachers. In *SIGCSE*, 2007.

[4] A. Bundy. Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1:67–69, 2007.

[5] Committee for the Workshops on Computational Thinking. *Report of a Workshop on The Scope and Nature of Computational Thinking.* The National Academies Press, 2010.

[6] D. D. Garcia, C. M. Lewis, J. P. Dougherty, and M. C. Jadud. If ________, you might be a computational thinker! In *SIGCSE*, 2010.

[7] P. B. Henderson, T. J. Cortina, and J. M. Wing. Computational thinking. In *SIGCSE*, 2007.

[8] G. Lewandowski, D. Bouvier, R. McCartney, K. Sanders, and B. Simon. Commonsense computing (episode 3): Concurrency and concert tickets. In *ICER '07: Proceedings of the Third International Workshop on Computing Education Research*, 2007.

[9] J. A. Qualls and L. B. Sherrell. Why computational thinking should be integrated into the curriculum. *Journal of Comp. Sci. in Colleges*, 25:66–71, 2010.

[10] J. Wing. Computational thinking. *Communications of the ACM*, 49:33–35, 2006.

APPENDIX

A. OUTLINE OF THE CT MODULE

Lecture 1: Computational Thinking and 21st Century Problem Solving

1. Object lesson: driving directions
   - "How do you get from school to the mall?" Clicker questions 1–2
   - "How did you think about the problem?" CT concepts: *Algorithm, Efficiency*
   - "What if the main road was closed?" CT concepts: *Debugging*
   - "How do computers solve this problem?" CT concepts: *Abstraction, Automation*
2. What is computational thinking?
   - An approach to *problem solving* which uses abstraction to create *algorithmic solutions* which can be automated with computation.
   - A *fundamental skill* used by everyone by the middle of the 21st century (i.e., just like reading, writing, and arithmetic).
3. Daily examples of CT
   - Looking up names in a phone book
   - Buying movie tickets (multiple lines)
   - Clicker question 3
   - CT is . . . , CT is *not* . . .
4. Concept #1: Abstraction
   - "CT is reformulating a seemingly difficult problem into one we know how to solve."
   - People standing in line → *Queue*
   - Cafeteria plates → *Stack*
5. Concept #2: Logical Thinking
   - Inductive reasoning: observation → pattern → hypothesis → theory
   - Deductive reasoning: theory → hypothesis → observation → confirm
6. Concept #3: Algorithms
   - Activity: "peanut butter and jelly sandwich"
   - Explain "write it/ do it" (Science Olympiad)
   - Homework: "swap puzzle" on cs4fn.org
7. Concept #4: Debugging
   - Clicker questions 4–5
   - Discuss reasoning behind each answer

Lecture 2: Computational Thinking in K-12

1. Review of previous lecture
   - Clicker questions 1–3
   - CT Concepts: decomposition, abstraction, logical thinking, algorithms, debugging, automation
   - More definitions for CT
2. Why is CT important for K-12?
   - Enhances problem solving techniques
   - Moves students beyond technology literacy
   - CT is a higher-level cognitive process
3. CT and problem solving strategies
   - *Heuristic*: an experience-based strategy that facilitates problem solving
   - *Algorithm*: a specific sequence of steps that guarantees a solution
4. How do you teach algorithms?
   - Demonstrate specific procedures; apply examples
   - Help students explain their thinking and debug
   - Role play: Towers of Hanoi
5. Using technology to motivate CT
   - Facebook friend network visualization
   - Google's public data explorer
   - Amazon/Netflix/etc recommendations
   - "What other tools have you seen?"
6. Applying CT to any content area
   - Science, Social Studies, Economics, . . .
   - Social Sciences, Medicine, Humanities, . . .
7. The big picture
   - CT is a fundamental skill for *everybody*
   - State of computing in secondary education
   - Highlight "CS Principles" proposed AP course

B. CLICKER QUESTIONS

Lecture 1 (Intro to CT)

1. How extensive were your directions?

[A] One step (e.g., type "mall" into GPS or Google Maps)
[B] Two steps (e.g., from downtown take bus #4)
[C] Several steps (e.g., head east on SR-15 to 3rd Street)
[D] A detailed, turn-by-turn route (e.g., from the union)
[E] None of the above (e.g., "Man, I was way off!")

2. How did you figure out the driving directions?

[A] Knew them already; simply "recalled" the route
[B] Sketched out a high-level map on paper
[C] Thought about several ways, picked one
[D] Texted a friend when no one was looking
[E] Modeled the entire city as an undirected graph, solved the "single-pair shortest path problem," and applied it to the source and destination

3. What is the quickest way to serve 20 pizzas to 60 hungry students?

[A] One table with pizzas (the usual case)

[B] Five tables with four pizzas each

[C] People stay put and pizzas are passed around

[D] Four servers bringing the pizza around

Scenario: You come home and the desk lamp in your apartment stopped working (it worked in the morning).

4. What is your first step to solve the problem?

[A] Check if the lamp is turned on

[B] Check if the light bulb is working

[C] Check if the lamp is plugged in

[D] Check if the outlet is working

[E] Check if there is power in the room

Scenario: You checked A-E and it is still not working.

5: What do you do next?

[A] Buy a new lamp

[B] Call your mother/friend/landlord/etc

[C] Use your roommate's lamp

[D] Repeat steps A-E from before

[E] Forget about the problem for the day

Lecture 2 (Review of CT)

1. What are the two main ideas of computational thinking?

[A] Abstraction and Automation

[B] Algorithm and Analysis

[C] Debugging and Logical Thinking

[D] All of the above

2. Computational thinking relies on the use of computer programs.

[A] True

[B] False

3. Computational thinking mainly involves computer science and has little impact on other subject areas.

[A] True

[B] False

C. POST-SURVEY

Indicate whether you (1) strongly agree, (2) agree, (3) disagree, or (4) strongly disagree.

1. Knowledge of computing will allow me to secure a better job.
2. My career goals do not require that I learn computing skills.
3. I doubt that I can solve problems by using computer applications.
4. I expect to use software in my future educational and career work.
5. I can achieve good grades (C or better) in computing courses.
6. The challenge of solving problems using computer science appeals to me.
7. I expect to use computer applications for future projects involving teamwork.
8. I can learn to understand computing concepts.
9. I am not comfortable with learning computing concepts.
10. I expect to use computing skills in my daily life.
11. I hope that my future career will require the use of computing concepts.
12. I think that computer science is interesting.
13. I will voluntarily take computing courses if I were given the opportunity.
14. Computational thinking can be integrated into classroom education in other fields.
15. Computational thinking should be integrated into classroom education for other disciplines.
16. Having background knowledge and understanding of computer science is valuable in and of itself.

Open-ended questions

1. In your view, what is computing? What is its purpose?
2. In your view, what is computational thinking?
3. How can we integrate computational thinking in the classroom?
4. How does computational thinking relate to other disciplines and fields? Please provide specific examples.