An Informatics Perspective on Computational Thinking

James Walden, Maureen Doyle
Dept. of Computer Science
Northern Kentucky University
Nunn Drive
Highland Heights, KY 41076
waldenj@nku.edu,
doylem3@nku.edu

Rudy Garns
Dept. of Philosophy
Northern Kentucky University
Nunn Drive
Highland Heights, KY 41076
garns@nku.edu

Zachary Hart
Dept. of Communication
Northern Kentucky University
Nunn Drive
Highland Heights, KY 41076
hartz@nku.edu

ABSTRACT

In this paper, we examine computational thinking and its connections to critical thinking from the perspective of informatics. We developed an introductory course for students in our College of Informatics, which includes majors ranging from journalism to computer science. The course covered a set of principles of informatics, using both lectures and active learning sessions designed to develop informatics and computational thinking skills. The set of principles was drawn from a wide set of sources, and included broad principles like those of Denning and Loidl, as well as more limited principles related to topics like universal computation and undecidability. We evaluated the change in both computational and critical thinking skills over the course of the semester, using a well-known validated critical thinking test and a computational thinking test of our own devising.

Categories and Subject Descriptors

K.3.2 [**Computing Milieux**]: Computers and Education-Computer and Information Science EducationCurriculum

General Terms

Active Learning, Non-majors, New Curriculum

Keywords

computational thinking, informatics

1. INTRODUCTION

The College of Informatics at Northern Kentucky University is comprised of three departments: Computer Science, Communication, and Business Informatics, also known as Information Systems, in which 2200 students are enrolled, 668 of which are in the Department of Computer Science. In order to provide students with a common foundation in informatics, we developed a new course, Principles of Informatics, for incoming freshmen. Many of the principles of informatics taught in the course are principles of computer science or computational thinking. We have offered this class since 2010, and the class will be offered as a pilot general education course in spring 2013. Offering Principles of Informatics as a general education course has several advantages, including reaching a wider range of students and enabling majors within the College of Informatics to add it as a required class without exceeding the maximum number of credit hours permitted.

Learning objectives for the course are:

1. Define and identify terms, concepts, and current practice of informatics.

2. Find, interpret, and evaluate information, including the assumptions, evidence, and theories behind it using multiple techniques and tools.

3. Evaluate the capabilities of information communication technologies based on an understanding of the scientific principles of informatics, such as the existence and limits of universal digital computers.

4. Explain how the design of information communication technologies such as the Internet influence human behavior.

5. Understand how information communication technology influences the creation of shared meaning.

Computational thinking is a form of discipline-specific thinking like mathematical thinking [16] or engineering design thinking [8]. It differs from other types of thinking discussed in the educational literature, such as critical thinking, in its tight disciplinary focus. While the goal stated in Wing's germinal paper [21] is to add computational thinking to the basic skills of reading, writing, and arithmetic, the definition of computational thinking may be too narrow and too tightly coupled to traditional definitions of computer science [15] to be widely accepted outside the discipline.

In this paper, we take a broad look at computational thinking from the perspective of informatics. The College of Informatics at the authors' university includes a broader set of majors than most, including not only traditional disciplines like Computer Science, Information Systems, and

ITiCSE'13, July 1-3, 2013, Canterbury, England, UK

Information Technology, but also Communication and Journalism as well as fields such as Health Informatics and Media Informatics. An NSF grant supported "informaticists in residence", selected from both faculty outside the College of Informatics and faculty outside the university, in order to bring knowledge and expertise from fields as diverse as history, philosophy, biology, and music.

The field of informatics focuses on the study of information in a broad sense, not limited to the processing of information. Informatics includes the objective, precise analysis of information found in Shannon's theory, the meaning of information, and the impact of information on behavior and society. Weaver calls these the technical, semantic, and effectiveness problems of information in his introduction to Claude Shannon's *The Mathematical Theory of Communication* [20]. While algorithms and processing data are an important part of studying information, the meaning of information and the impact of information on organizational and societal issues, such as communication, security, and privacy are also essential to a broad understanding of information.

While many see the value of teaching computational thinking as part of general education requirements [5][6], Principles of Informatics is not a programming course nor a course focused solely on computational thinking, but a computer literacy course for the 21st century. The signficant contribution to computer science education presented in this paper is not only to spur discussion about computing literacy but also to demonstrate that computational thinking must be taught to a broader pool of students and that this can be accomplished via a single transdisciplinary course. We discuss the transciplinary course, creation of evaluation instruments, and share the course curriculum via the course web site at `http://inf128.nku.edu/`.

There has been success infusing courses with computational thinking and computer science principles [14][19][2], while others have taught CS Principles courses in high schools and universities [1]. The Principle of Informatics course introduces some CS principles and computational thinking skills, but it is not intended to be a CS Principles course or computational thinking course. It is aimed at the broader field of informatics, to which computer science belongs, but which also includes material from communication, information systems, library science, and other information focused disciplines.

2. COURSE DEVELOPMENT

The course development team consists of three computer scientists, a philospher, a historian, a biologist, an associate professor of communication, and a professor of mathematics education doing evaluation. The group previously included personnel from from the fields of information systems, music, and social informatics.

2.1 History

The course has been taught three times, including the fall 2012 semester. The major concepts of the course and its structure have not changed. The class meets twice a week. The first meeting is a lecture day, during which the entire class meets together, while the second meeting is an active learning day, in which the class meets in sections of approximately 20 students each. Students are divided into small groups for the active learning sessions. These groups are also the same groups that work together on a semester long project. The course is team taught by instructors from different disciplines, along with a group of guest lecturers that typically come from the course development group.

2.1.1 Fall 2010

Sixty students enrolled in the first offering of the class held in fall 2010, which was team taught by three instructors, one from each of the departments in the College of Informatics. Lectures were given by one of the team instructors or by guest lecturers from the course development group, while each instructor had a section of 20 students for the active learning sessions.

2.1.2 Fall 2011

With less excitement about a second offering from advisors and students, only eleven students registered for the course in fall 2011. The course was taught by a professor from the department of communication. Similarly to the previous year, lectures were delivered by instructors who created the content, and most of the students were majoring in computer science or information technology.

2.2 Current Offering

A new format for Principles of Informatics course substantially increased enrollment to 47 students during the fall 2012 semester. The class was part of a learning community with a public speaking class. Students who registered for a section of the Principles of Informatics class were also registered in a corresponding section of a public speaking class. A computer science professor taught one section of the Principles of Informatics class, while a communication professor taught the other section. They collaborated with the public speaking instructors to identify informatics related topics and issues that could be covered in the speech assignments in the public speaking class. Approximately half the students continued to be CS or IT majors, while the remaining students majored in other areas of informatics, including undeclared informatics majors, representing a greater diversity of students than had been previously enrolled.

The two sections of the Principles of Informatics class met one day a week in the Griffin Hall Digitorium, a high tech auditorium with a 16-foot tall video wall, for interactive lectures. These lectures included the use of a Twitter backchannel that was displayed on the video wall for questions and discussion. Additional video and graphical elements were displayed on the screen to enhance student understanding. Each student received an iPad for the semester to use for tweeting questions and comments throughout the week and to participate in learning activities during the active learning sessions.

3. PRINCIPLES OF INFORMATICS

Paralleling the development of the ideas of algorithmic or computational thinking skills, sets of principles or fundamental ideas to define computer science and informatics have been suggested and debated since at least 1980 [18]. While our work was inspired by earlier efforts such as Denning's great principles of computing [7] and Loidl et. al's principles of informatics [17], we wanted to produce a larger list of both high level ideas and more specific principles along the lines of how a science like physics has a catalog of both general and specific laws, principles, and theories.

The set of principles we developed for this class were divided into the following categories: data and information, business, communications, computation, networking, and security and privacy. These categories are similar to the windows into which Denning divided his great principles of computing [7] though they share the breadth of Loidl et. al's basic concepts of informatics. Within a category are more specific principles, such as this set of principles for information:

1. Information can be measured using the Shannon definition.
2. Evolution is a process for selecting information.
3. Digital information consists of discrete binary units called bits.
4. Information can be encoded into bits in many ways.
5. Digital copies are easy to make and perfect, while analog copies degrade with each generation.
6. Metadata consists of bits describing other bits.
7. Bits can last forever, but you may not be able to decode them.

The principles were drawn from a wide variety of sources, including well established principles like Kerchoff's Principle in cryptology or Amdahl's Law in parallel computing as well as newly stated principles from sources like Denning [7] and Loidl [17] along with modern coursework like Blown to Bits [3] and from our own work. There was no textbook for the course. Instead of a textbook, readings were either selected from external sources or were developed specifically for the principles being studied.

While the full list of principles cannot be covered in detail in a single course, our goal is to integrate them into the curricula of the disciplines in the College of Informatics and to offer an introduction to a core set of principles in our introductory class.

4. CRITICAL THINKING

Like computational thinking, critical thinking is defined in many ways, often with less than desired precision. Most generally, we can describe critical thinking as a kind of thinking that is concerned with the evaluation of judgments and arguments, particularly with regards to truth or credibility. Appealing to what may be closest to expert consensus, the APA (American Philosophical Association) understands critical thinking to be "purposeful, self-regulatory judgment which results in interpretation, analysis, evaluation, and inference." [10]

Any definition of critical thinking must highlight its normative or evaluative dimension. We want to distinguish good arguments from bad arguments and weak evidence from strong evidence; concerns for truth and credibility guide these endeavors. As well we need to assess relevance, understand context and appeal to recognized, well-established epistemological norms. To do this well requires an understanding of the formal and procedural demands of logic and sound argumentation. Critical thinkers need to comprehend the concepts of validity and a well-formed formula, and they must be proficient in their abilities to construct and recognize the structure of arguments, apply rules, generalize, abstract, create models and consider counter-examples.

We noticed an apparent overlap between critical thinking and recent characterisations of computational thinking, so we proposed to explore the relationship, looking to see whether our course emphasizing principles of informatics would lead to growth in critical thinking skills and whether growth in computational thinking would be correlated with growth in critical thinking. There are few widely available instruments designed specially to measure growth in computational thinking; there are, however, numerous well-known instruments available to measure growth in critical thinking [9].

Both critical thinking and computational thinking lend themselves to a desire to distribute skills and attitudes across the curriculum; they contribute importantly to cultivating information-age citizens, and they focus on skills and abilities that, at a formal and procedural level, have much in common.

Having identified common ground between computational and critical thinking, we must note important differences. We can see the differences when we compare the acts of critiquing and computing. The former has a normative dimension that is not present in the latter. Thinking critically, as an ultimately evaluative activity would involve finding and evaluating evidence, questioning assumptions and assessing premises, avoiding biases, and checking the credibility and reliability of sources.

Computational thinking, on the other hand, is primarily an activity that has a formal, procedural orientation, emphasizing processes and strategies for problem solving. Computational thinking includes elements from mathematical thinking, particularly constructive solutions and algorithms, and from engineering design thinking [22], while also including novel elements, such as the performance and scalability of solutions. It is in the engineering design thinking aspects of computer science that it most closely approaches the focus of critical thinking on evaluation.

While critical thinking tends to be defined in terms from the humanities, definitions of computational thinking typically borrow heavily from the metaphors and the conceptual lexicon of computer science, and all too often it is framed as a kind of cognitive activity designed to capture how computer scientists think [15]. Characterizations of critical thinking, on the other hand, more easily abstract from specific disciplines across the humanities and social sciences. The movement to teach critical thinking across the curriculum has a long established and fairly successful history while computational thinking is a new movement not widely known outside of computer science. Opportunities to learn and practice computational thinking provide a rich set of contexts and examples for learning to be come a better critical thinker.

5. PRELIMINARY EVALUATION

As instructors, we are interested in evaluating whether students' computational thinking and critical thinking skills improve as a result of taking the Principle of Informatics course. We are also interested in examining the relationship between computational thinking and critical thinking. One way to study this is to examine if there are strong correlations or other relationships based on student performance on pre and post exams administered at the beginning and

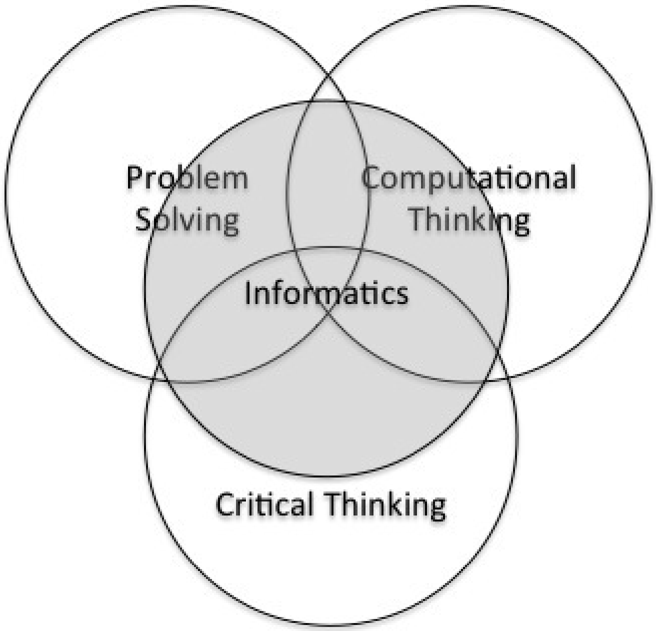

Figure 1: Intersection of Computational and Critical Thinking

end of the term in each area. We developed our own computational thinking instrument, while we used commercially available critical thinking tests.

We also evaluated the effectiveness of this course to teach computational thinking skills to a general audience. These results are preliminary and evaluation continues.

5.1 Critical Thinking

The Critical Thinking Co. Z-test is a 50 minute online test designed to evaluate a students use of induction, deduction, credibility, assumptions, and meanings for making decisions about an action or belief. It employs a multiple choice format. The instrument has been validated and shown to be reliable. Students were encouraged, but not required, to take the exam.

In fall 2010, 25 students took both the pre and post critical thinking tests. While the average grade improved over the term, the results were not statistically significant. Students earned an average of 23.86 out of a possible total of 52 questions, with a standard deviation of 14.6, on the initial test. The post test average score was 27.8, with a standard deviation of 16.53. Sixteen (65%) of the student scores improved. A paired Student's t-test compared the pre and post tests, but the results were not significant (p=0.24). Additionally, 36% of the students performed worse in the final test than the initial test.

In 2012 we administered the California Critical Thinking Skills Test [11] which is also an online, 50 minute test. Seventy students took some part of this exam, but student's results for this test, and its complementary computational thinking test, were eliminated when they spent fewer than 15 minutes on the timed test or missed the pretest. After elimininating students who did not spend time on the exam, the pretest was taken by 62 students, 43 enrolled in the Principles course and 19 in the control group. Thirty-three students took both the pretest, posttest and took at least 15 minutes on both critical thinking tests. All but two of these students were enrolled in the Principles course. Figure 2 shows the pre and post paired exam results for students in the control group (left hand side, 'No') and in the Principles course (right hand side, 'Yes'). The pre test results are blue or dark boxes, while the post test results are shown with cream or light-gray boxes. The width of each box indicates the relative number of students who took each exam.

A paired Student's t-test was performed. There were no significant differences in critical thinking skills between the pre test and post test for the aggregate group of students from all classes, the control group, or students enrolled in the Principles of Informatics class.

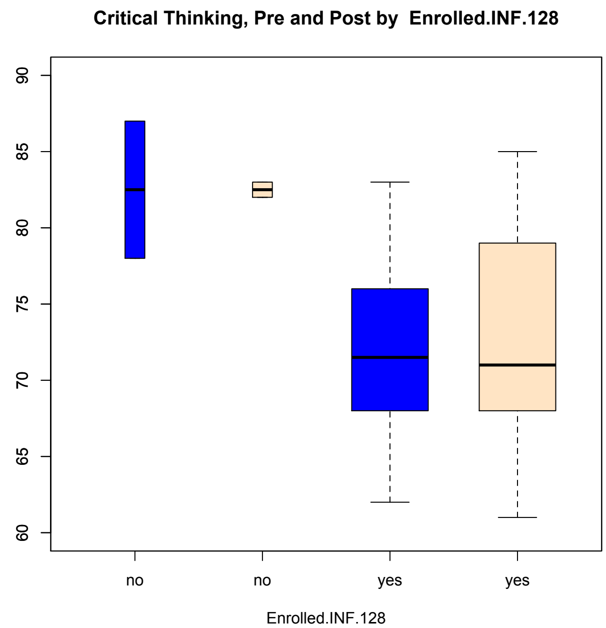

Figure 2: Critical Thinking pre and post results

Fall	Total	Improvement	p-value
2010	29	No	0.3935
2011	7	Yes	0.017
2012	31	Yes	<0.0001

Table 1: Computational Thinking Improvements

5.2 Computational Thinking

Computational thinking exams were given to students enrolled in INF 128 in fall 2010-2012 to measure if the class improved students computational thinking skills. Students took the exam twice, once during the first week of class and a second time during the last week of class. A paired Student's t-test was performed for each semesters' data for all students entrolled in the Principles course who took both the pre-test and the post-test. Table 1 summarizes the exam results.

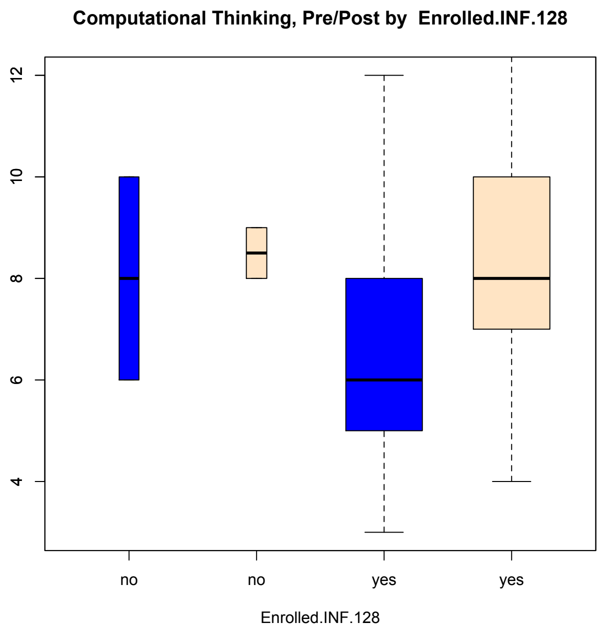

Figure 3: Pre and Post Computational Thinking Scores

The exam tests evaluation of simple algorithms, efficient sorting, quality of digital information storage, and file structures. The test was a combination of multiple choice and short answer questions.

In fall 2012, a control group participated in the test taking. Students were awarded extra credit to participate. Figure 3 is a box plot, showing the pre and post test performance on the Computational Thinking test. The same format was used for this graph as for figure 2.

A paired t-test was performed to see if there was a difference in means for students between the pre and post exam. Students enrolled in the class showed signficant improvement in the computational thinking scores ($p < 0.0001$), while students in the control group could not reject the hypothesis that the pre and post test results were the same ($p = 0.87$).

An unpaired t-test was performed to test if there was any signficant difference between exam results for the control group versus the students in the class. No signficant difference in the pre or post exam between the control group and the students enrolled in the class could be shown. This is most likely due to the low turnout for the control group and additional effort will be made to recruit more students for both the pre and post test in future iterations of the class.

5.3 Critical and Computational Comparisons

No correlation was found between the critical thinking and computational thinking scores in 2010. We did not administer a critical thinking test in fall 2011 because of low enrollment.

If critical thinking and computational thinking measured closely related skills, then a significant improvement in one should result in a signficant improvement in the other. This is not the case, as computational thinking signficantly improved but critical thinking did not for students enrolled in the Principles class.

Another indicator of similarity would be strong correlations between the differences in the pre and post critical thinking and computational thinking scores for students who took all four exams. The Pearson's correlation coefficient $\rho = -0.09$ with a p-value of 0.6 indicates there was no correlation in the differences between the two tests. More data is needed to be conclusive.

We are also interested in how the two skill sets are similar. We examined the student scores between the Critical Thinking and Computational Thinking scores for the pre-test and post-test. There is a strong correlation between both sets of data, the pre-test has a $\rho = 0.41$ with $p = 0.008$, which is significant, and the post-test with a $\rho = 0.41$ with $p = 0.02$. This indicates that about 15-20% of the variablity in one test can be explained by the other.

5.4 General Education

The Principles of Informatics course will be offered as a general education course in spring 2013, which should attract a wider range of majors to the class. For fall 2012, we checked our data for signficant differences between the test scores of Science, Technology, Engineering and Mathematics (STEM) students and non-STEM students. Seventeen of the 33 students who took the pre and post tests were STEM majors. The only signficant difference in performance was in the pre computational thinking test with $p = 0.016$. STEM majors results were higher than non-STEM majors. There was no signficant differences between students who were in the College of Informatics ($n = 31$) and those who were not.

5.5 Future Evaluations

In fall 2012, we used a new critical thinking test. There were various technical difficulties with the Critical Thinking Co. Z-test, so we began using Insight Assessment's California Critical Thinking Skills Test [11]. This instrument is valid and reliable. There is also excellent test administrator support. Testing will continue while funding is available for this test. The cost is currently $10 per exam. Evaluation of computational thinking will continue for each course offering at our institution and will be modified to improve its reliability.

The use of a control group will continue whenever the Principles of Informatics course is part of a learning community, as it was in the 2012-2013 academic year. To encourage more participation in this study, students will be awarded extra credit to take the pre and post exams. In addition, students who complete both the pre and post tests and spend more than 15 minutes on them will be entered into a drawing for a tablet.

The Principle of Informatics course will be offered at another institution in fall 2013. The pre and post exams will be given at this institution to determine the reproducability of the results.

5.6 Threats to Validity

The use of the same tests for pre test and post test is a threat to validity since students may remember the questions and look at the answers. Students volunteer to take the exam and this introduces a threat that these results may not be reflective of all students enrolled in the class; similarly, the results may not extend beyond the school currently teaching the course. Both critical thinking tests The Critical Thinking test, created by Critical Thinking Inc. [9] and the California Critical Thinking Test [4], [12] are a valid and reliable tests. The Computational Thinking test is still in development and is not verified valid or reliable instrument.

6. CONCLUSIONS

In this paper, we described the nature and evaluation of a new transdisciplinary freshman level course, Principles of Informatics. The support of an NSF grant enabled us to bring in outside experts from a diverse set of fields as "informaticists in residence" to help us develop the course. In developing and teaching the course, we examined computational thinking from the perspective of informatics, including how computational thinking may be related to critical thinking, with the help of our informaticists in residence. We used longitudinal studies of both computational and critical thinking to evaluate how the course impacted the skills of our students in both types of thinking. Our evaluations included a new computational thinking instrument that we developed and a well known validated critical thinking test.

7. ACKNOWLEDGMENTS

The authors would like to acknowledge the support of the NSF CPATH program grant 0939103. We would like to acknowledge helpful conservations with Chris Curran, Gina Folleta, Kevin Kirby and Bill Landon.

8. REFERENCES

[1] Astrachan, O., Barnes, T., Garcia, D.D., Paul, J., Simon, B., and Snyder, L.. 2011. CS principles: piloting a new course at national scale. In Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE '11). ACM, New York, NY, USA, 397-398.

[2] Barr, V., Liew, C.W., and Salter, R.. 2010. Building bridges to other departments: three strategies. In Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10). ACM, New York, NY, USA, 64-65.

[3] Hal Abelson, Ken Ledeen, and Harry Lewis. 2008. *Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion*. Addison-Wesley Professional.

[4] Bondy, K.N., Koenisgseder, L.A., Ishee, J.H., Williams, B.G., Psychometric properties of the California Critical Thinking Test, J Nurs Meas 2001 Winter; 9(3), pp. 309-328.

[5] Curzon, P., Peckham, J., Taylor, H., Settle, A., Roberts, E., 'Computational Thinking (cT): On Weaving It In', Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education, ACM, New York, USA, pp. 201-202.

[6] Dierbach, C., Hochheiser, H, Collins, S, Jerome, G., Ariza C., Kelleher T., Kleinsasser W., Dehlinger, J., Kaza S., 'A model for piloting pathways for computational thinking in a general education curriculum,' Proceedings of the 42nd ACM technical symposium on Computer science education (sGICSE '11). ACM, New York, USA, pp. 257-262.

[7] Peter J. Denning. 2003. Great principles of computing. Commun. ACM 46, 11 (November 2003), 15-20.

[8] Dym C. et. al. Engineering Design Thinking, Teaching, and Learning, Journal of Engineering Education, January, 2005, 103-120.

[9] Ennis, R.H., Millman, J., Tomko, T.N., "Administration Manual for Cornell Critical Thinking Tests Level X and Level Z, 5th Edition, Revised," The Critical Thinking Co., pp. 16,18-32.

[10] The APA Delphi Report, Critical Thinking: A Statement of Expert Consensus for Purposes of Educational Assessment and Instruction 1990 ERIC Doc. NO.: ED 315 423

[11] Facione, P.A. and Facione, N.C. (1992) The California Critical Thinking Disposition Inventory Test Manual. The California Academic Press, Millbrae, CA.

[12] Facione, P.A. (1990) The California Critical Thinking Skills Test: College Level Technical Report #1 - Experimental Validation and Content Validity. The California Academic Press, Millbrae, CA.

[13] Fisher, A. and Scriven, M. (1997). *Critical thinking: its definition and assessment*. Edgepress.

[14] Hambrusch, S., Christoph Hoffmann C., Korb J.T., Haugan, M., and Hosking, A.L. 2009. A multidisciplinary approach towards computational thinking for science majors. In Proceedings of the 40th ACM technical symposium on Computer science education (SIGCSE '09). ACM, New York, NY, USA, 183-187.

[15] David Hemmendinger, A plea for modesty, ACM Inroads, v.1 n.2, June 2010

[16] Knuth, D. Algorithmic Thinking and Mathematical Thinking, The American Mathematical Monthly, Vol. 92, No. 3. (Mar. 1985), 170-181.

[17] Loidl, S., Muhlbacher, J., and Schauer, H. 2005. Preparatory knowledge: Propaedeutic in informatics. In From Computer Literacy to Informatics Fundamentals, R. T. Mittermeir, Ed. Springer, Berlin, Germany, 105-115.

[18] Niegervelt, J. 1980. Computer science education: An emerging consensus on basic concepts. In Information Processing 80, S. H. Lavington, Ed. North Holland, Amsterdam, 927-933.

[19] Perkovi, ï£¡, P., Settle, A., Hwang, S., and Jones, J. 2010. A framework for computational thinking across the curriculum. In Proceedings of the fifteenth annual conference on Innovation and technology in computer science education (ITiCSE '10). ACM, New York, NY, USA, 123-127.

[20] Claude E. Shannon and Warren Weaver. 1963. *A Mathematical Theory of Communication*. University of Illinois Press, Champaign, IL, USA.

[21] Jeannette M. Wing, Computational thinking, Communications of the ACM, v.49 n.3, March 2006.

[22] Jeannette M. Wing, Research Notebook: Computational Thinking–What and Why? `http://link.cs.cmu.edu/article.php?a=600`, Spring 2011.