ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

A Conceptual Framework for Computational Thinking as a Pedagogical Device

Dr. Swati V Chande

Professor, Dept. of Computer Science, International School of Informatics and Management, Jaipur, Rajasthan, India

ABSTRACT: The permeation of computers in every aspect of human life has led to the recognition of the knowledge of computers as a necessity. People in every domain are looking for automation of their tasks and solutions to their problems through computers. It is therefore required that the generations to come are able to apply computational approach to problem solving early on and hence to incorporate not only knowledge of computers but also computational thinking in the curriculum at various stages in teaching-learning and a formal pedagogical device for it would lead to effective imparting of knowledge in this domain. This paper discusses the pedagogical aspects of Computational Thinking and drafts a pedagogical device for imparting computational thinking skills to the learners.

KEYWORDS: Computational Thinking, Pedagogy, Pedagogical device, Education, Problem solving

I. INTRODUCTION

Since computers have become an all-pervading tool to automate regular, repetitive tasks and solve complex, voluminous problems their use as a problem solving and automating tool cannot be overemphasized in any domain. All problem-solving thus seems to have become a computer programming exercise, and knowledge of Computer Science a basic skill.

The science that scientists and researchers developed drawing inspirations from natural processes now looks to be taking the center stage and reversely motivating them to decipher natural processes as computational activities. The complexities that the computer can handle, the speed and the accuracy of results act as inspirations to think of complex natural processes as computational tasks and researchers have been successful in identifying even the Computer Science behind Natural Selection by thinking computationally.

The idea of Computational thinking is that of a thought process involved in formulating problems and their solutions so that the solutions are represented in a form that can effectively be carried out by an information-processing agent. [1]

Computational thinking (CT) enables bending computation to one's needs and is becoming the new literacy of the 21st century. A structured pedagogical framework is therefore necessary to facilitate effective learning of Computational thinking. The 2011 workshop of the National Research Council on Pedagogical Aspects of Computational Thinking [2] has been a pioneering step in this direction and has raised several key points on the theme to trigger research and formulation of structures in this direction.

This paper is an early attempt to deliberate on the issue of inclusion of CT as a pedagogical device. Section 2 discusses the principles of computing and computational thinking section 3 views Computational Thinking from the pedagogical perspective, Section 4 provides a broad framework for the application of CT as a pedagogical device and section 5 concludes the paper while stressing the possibility for further research in this area.

II. PRINCIPLES OF COMPUTING AND COMPUTATIONAL THINKING

Computation is an expansive term that includes diverse tasks, concepts and techniques. Similarly, computational thinking involves an extensive set of approaches and skills, with applications in several disciplines.

Computational thinking (CT) is an expression coined by Jeannette Wing [3] to describe a set of thinking skills, habits and approaches that are integral to solving complex problems using a computer and widely applicable in the information society. CT involves defining, understanding, and solving problems, reasoning at multiple levels of abstraction, understanding and applying automation, and analyzing the appropriateness of the abstractions made. It shares elements with various other types of thinking such as algorithmic thinking, engineering thinking, design

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

thinking, and mathematical thinking and as such, draws on a rich legacy of related frameworks as it extends previous thinking skills [4].

These elements of computational thinking, common to different scientific thinking domains are represented in Figure 1.

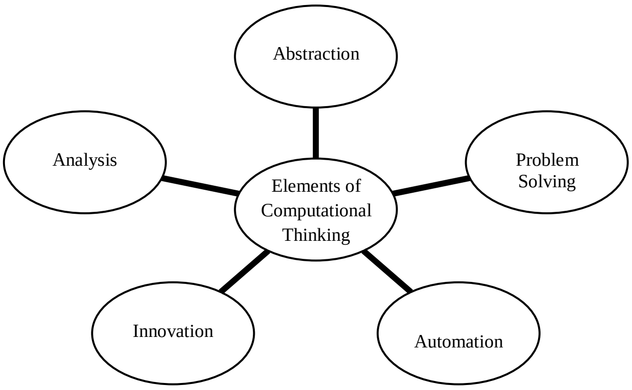

Figure 1: Elements of computational Thinking

The Great Principles of Computing, according to Denning [5] are: computation, communication, coordination, recollection, automation, evaluation, and design. These seven principles form a foundation that is useful to recognize, organize, and categorize instances of computational thinking and build a framework that can translate computational thinking to contexts outside of Computer Science [6]. Figure 2 depicts these seven principles.

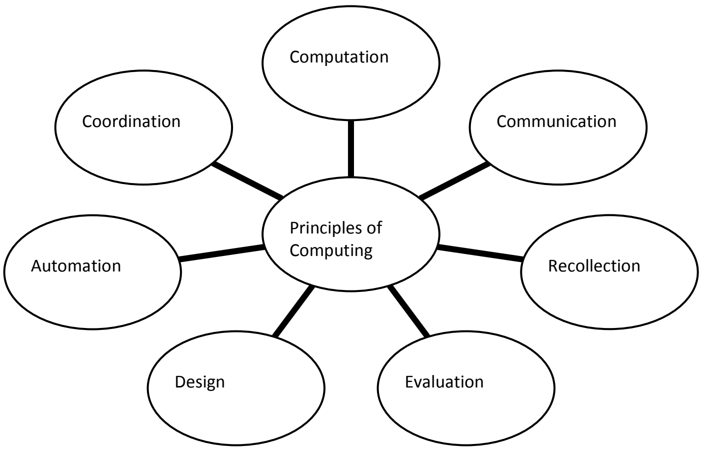

Figure 2: Principles of Computing

The principles of computing find application in computational thinking, and computational thinking finds applications in solving complex problems in several disciplines including in contexts outside of Computer Science. As mentioned in [1], Computational thinking for everyone means being able to:

- Understand which aspects of a problem are amenable to computation,
- Evaluate the match between computational tools and techniques and a problem,
- Understand the limitations and power of computational tools and techniques,
- Apply or adapt a computational tool or technique to a new use,
- Recognize an opportunity to use computation in a new way, and
- Apply computational strategies such as divide and conquer in any domain.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

It is to aid the development of computational thinking skills that a pedagogy interweaving the above functions with the principles of computational thinking is necessitated.

III. COMPUTATIONAL THINKING: A PEDAGOGICAL PERSPECTIVE

Development of thinking skills has been identified as a necessity for classroom teaching to ensure that the students do not merely turn out to be data repositories but are able to make effective and intelligent use of the data or information at hand. Mike Fleetham [7] has set forth the necessity of imparting thinking skills in order to meet the meet the demands of the evolving world that needs more and more problem solvers, decision makers and innovators.

A pedagogic device is described by Bernstein as the ensemble of rules or procedures via which knowledge is converted into classroom talk, curricula and online communication [8]. Bernstein [9] identified three main fields of the pedagogic device, namely, the field of production, recontextualization, and reproduction. These fields are hierarchically related, in that, recontextualization of knowledge cannot take place without the original production of knowledge, and reproduction cannot take place without recontextualization. Recontextualisation is a process that extracts text, signs or meaning from its original context in order to establish it into a different context [10].

Bernstein's ontological position can be interpreted in a number of ways [11], and the following interpretation is one such version.

In the context of Computational Thinking, the original production of knowledge is consequent from Computer Science, it is the recontextualization field that CT needs to focus so as to facilitate development of Computational Thinking as a pedagogical tool. The idea is to introduce thought processes that parallel a computer's processing and enable students to be lifelong learners and incorporate computing tools.

The Knowledge Production field for Computational Thinking includes processes and competencies from Computer Science. The Recontextualization field involves integration of the computational theories, concepts and processes with fields in which these could be applied i.e. extraction of text, symbols, concepts, theory and meaning from Computer Science to introduce it into another subject or context. The reproduction field includes specific instructional and regulative discussions at different levels and in different contexts.

IV. FRAMEWORK FOR COMPUTATIONAL THINKING AS A PEDAGOGICAL DEVICE

The authors in [12] have defined a conceptual framework to be 'a coherent system of concepts that flow from an objective'. To devise a conceptual framework for CT as a pedagogical device, the source objective is to ideate components that could provide a systematic direction to acquire and apply CT skills to different problems. While incorporating CT as a pedagogical device, as stated in [13] the objective is to discover better ways of envisioning the potential for CT across all disciplines and find ways to facilitate learning across and between these disciplines so that students can be provided with the skills they need to solve current and future challenges. The need is to help computer science professionals demystify their terminology to make it more widely understandable and inclusive, help teachers find ways to integrate CT knowledge/skills with their current knowledge and practice, present ideas and materials in terms/contexts that teachers will be comfortable with, and to communicate with educators using the language/terminology that is common in the educational environment. Studies such as [14][15][16][17][18][19][20][21] on application of CT as a didactic instrument and on planning the framework for CT as a pedagogical equipment reflect steps in this direction.

Computational thinking facilitates new ways of seeing existing problems. It does this by emphasizing creating knowledge rather than using information, presenting possibilities for creatively solving problems [22] and facilitating innovation. Computational thinking has also been described as the use of abstraction, automation, and analysis in problem-solving [1].

With the above objectives and concepts, an attempt is made to frame a conceptual pedagogy around the aspects of Computational thinking described in [1],

Imparting Knowledge of fundamental concepts and principles of Computer Science, types of computational problems, computational tools and techniques, problem solving, and abstraction is a pre-requisite.

- Identify patterns in the problem, and divide the problem into sub-problems then find out which of these could be solved using computational techniques. (*Identification*).
    - Learners should have knowledge of computational techniques and where these could be applied.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

- Evaluate the match between computational tools and techniques and a problem. (*Abstraction*).
    - Learners should be able to map the problem to be solved and the techniques to be applied in the context of the problem at hand.
- Understand the limitations and power of computational tools and techniques. (*Analysis*).
    - Students should be able to find out to what extent a problem could be solved using the computational techniques/tools.
- Apply or adapt a computational tool or technique. (*Automation*).
    - Scholars should be able to solve the problem using computational tools/techniques and computerize the solution to the problem.
- Apply or adapt a computational tool or technique to a new use, and Recognize an opportunity to use computation in a new way. (*Innovation*).
    - After acquisition of the above skills, the candidates could be expected to be able to apply computational tools and techniques to new problems.

An illustration of the pedagogical device for computational thinking (PDCT) is given in figure 5.

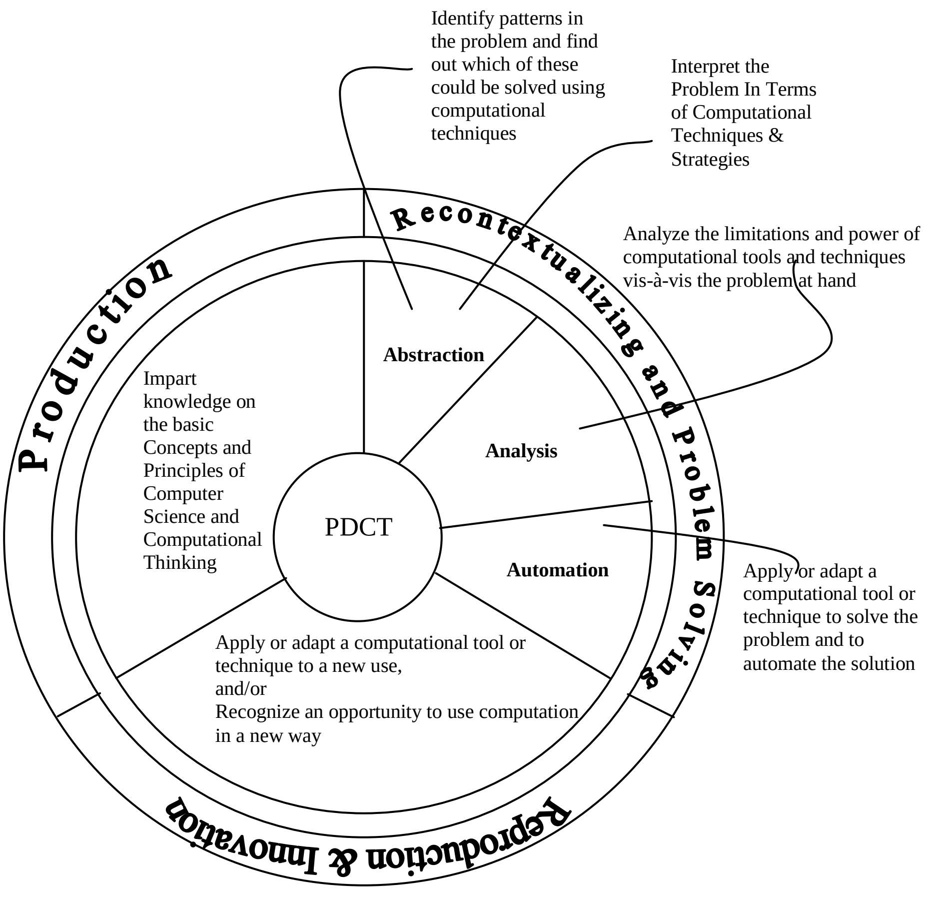

Figure 3: A pedagogical device for Computational Thinking

V. AN ILLUSTRATION OF APPLICATION OF THE DEVICE

The proposed pedagogical device is applied in the following text to a problem picked from the topics of the middle school curriculum of Indian schools. The operation of the proposed pedagogical device could be refined further to suite practical implementation.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Problem: Identifying herbs, shrubs and trees.

The students are expected to learn the characteristics of different types of plants and distinguish between them and then having understood the distinguishing criteria, they should be able to identify the varied plant types. They should be able to think computationally and apply computational techniques to automate the identification process.

ProductIon

Identification

The students, after having been made aware of the characteristics of herbs, shrubs and trees, and having been educated about basics of algorithms, decomposition and patterns could be taken to the school premises/garden to take notes on the foliage while applying computational thought

Recontextualization

Abstraction

Learners should be able to chart out the steps to be taken for systematic observation and recording of data, to break down the problem of identification of the foliage in to sub sections, and to spot the patterns in the vegetation.

Analysis

Students should be able to find out to what extent the identification of plant types could be done using the computational techniques. What criteria could be used to define the scope of identification as per the computational techniques.

Automation

Students should be able to identify the plant types using computational techniques and subsequently computerize the identification.

Reproduction

Reproduction and Innovation

After solving the plant type identification problem, the learners should be able to apply the computational techniques to new problems.

They could be given assignments to solve such problems independently.

Educating the learners about computational techniques and thinking computationally would equip them with a skill to find easy solutions to problems using computers. In [23], the researchers have proposed a framework for integration of computational thinking in the context of higher education and have illustrated examples of how computational thinking can be taught in a variety of Liberal Studies courses.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

VI. CONCLUSION AND FUTURE WORK

The objective of imparting Computational Thinking skills is to equip students with the ability to generate computational solutions to problems.

Computational thinking skill as per its advocates will have to be developed like other basic skills of Language and Mathematics. Attempts to formulate the pedagogy for integrating computational skills in school and college education are already on. It is still some time and research though, for it to become a full-fledged discipline. CT however has already been incorporated in the curriculum of some universities [24][25][26][27][28] and is recognized as an effective pedagogical device [2]. Formal structures for pedagogy for CT are needed to have CT recognized as a basic skill universally till it becomes an integral component of the subjects in which it finds it utility.

The proposed device is expected to act as a guideline for designing Computational Thinking instruction and is groundwork for outlining a structured approach to CT teaching and learning. Each of the components of the device could be further elaborated upon in the context of specific application domains. Use of computational perspective to edify a wide range of subjects has been a focus of study in academic circles, some of these include [29][30][31][32][33][34]. The incorporation of CT as a pedagogical tool would provide a formal base for building subject based teaching tools for problems that can be solved using computers.

REFERENCES

1. Cuny Jan, Snyder Larry and Wing Jeannette M, “Demystifying Computational Thinking for Non-Computer Scientists,” work in progress, 2010.
2. Committee for the Workshops on Computational Thinking, National Research Council ,Report of a Workshop on Pedagogical Aspects of Computational Thinking, National Academies Press, ISBN 9780309214742, August 2011.
3. J. Wing, Computational thinking, Communications of the ACM 49(3), pp. 33-35, 2006, DOI: 10.1145/1118178.1118215.
4. Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith and Linda Werner, Computational Thinking for Youth in Practice, ACM Inroads 2011 March, Vol. 2, No. 1, pp 32-37, DOI: 10.1145/1929887.1929902.
5. Denning Peter J, Great principles of computing, Communications of the ACM - Blueprint for the future of high-performance networking, Volume 46 Issue 11, November 2003, Pages 15-20.
6. Perkovi´c Ljubomir, Settle Amber, Computational thinking across the curriculum: A conceptual Framework, 2009. http://compthink.cs.depaul.edu/FinalFramework.pdf, Accessed October 12, 2015, DOI: 10.1145/1822090.1822126.
7. Mike Fleetham, How to Create and Develop a Thinking Classroom, LDA, 2003
8. Singh, Parlo, Pedagogising Knowledge: Bernstein’s Theory of the Pedagogic Device, British Journal of Sociology of Education, v.23, no.4, December 2002, p.571-582, DOI:10.1080/0142569022000038422.
9. B. Bernstein, Class, Codes and Control Vol.4: The Structuring of Pedagogic Discourse, Routledge, London, 1990.
10. http://edu.moca.org/education/teachers/curric/glossary, Accessed January 12, 2015.
11. Haugen Cecilie Rønning, Contextualizations and Recontextualizations of Discourses on Equity in Education, , Thesis for the degree of Philosophiae Doctor Trondheim Norwegian University of Science and Technology, November 2009., http://ntnu.diva-portal.org/smash/get/diva2:302950/FULLTEXT02, Accessed January 10, 2014.
12. Donald E. Kieso, Jerry J. Weygandt, Terry D. Warfield, Intermediate Accounting, Wiley, 14th Edition, 2011, http://www.wiley.com/legacy/college/kieso/0470374942/ifrs/ch02.pdf accessed on October 31, 2015.
13. The Computational Thinking Leadership Toolkit http://csta.acm.org/Curriculum/sub/CompThinking.html, Accessed January 10, 2014.
14. James J Lu, George H.L. Fletcher, Thinking about computational thinking, SIGCSE '09 Proceedings of the 40th ACM technical symposium on Computer science education, Pages 260-264, ACM New York, NY, USA 2009, DOI: 10.1145/1508865.1508959.
15. Enrico Pontelli, DIScovery SciEnce through Computational Thinking (DISSECT), http://www.cs.nmsu.edu/gk-12/wp-content/uploads/2011/09/GK12-Overview.pptx., 2011, Accessed December 17, 2014.
16. Wu He, “A Computational Thinking-oriented Online Lesson Planning System”, IEEE Technology and Engineering Education, ISSN 1558-7908, Volume 7, Number 1, pp 27-30, March 2012.
17. Kyu Han Koh, Ashok Basawapatna, Vicki Bennett, and Alexander Repenning, “Towards the Automatic Recognition of Computational Thinking for Adaptive Visual Language Learning” Proceedings of the 2010 Conference on Visual Languages and Human Centric Computing [VL/HCC 2010], IEEE Computer, pp 59-66, Madrid, Spain, DOI: 10.1109/VLHCC.2010.17.
18. Malyn-Smith Joyce and Lee Irene, “Application of the Occupational Analysis of Computational Thinking-Enabled STEM Professionals as a Program Assessment Tool”, Journal of Computational Science Education, Volume 3, Issue 1, pp 1-9, June 2012.
19. K P Soman, Manu Unni V.g, Praveen Krishnan and V Sowmya. Article: Enhancing Computational Thinking with Spreadsheet and Fractal Geometry: Part 1. International Journal of Computer Applications 55(14):1-8, October 2012. DOI: 10.5120/8820-2741.
20. Basawapatna, A., Koh, K. , H., Alexander Repenning, "Recognizing Computational Thinking Patterns", SIGCSE '11 Proceedings of the 42nd ACM technical symposium on Computer science education, Dallas, TX USA, DOI: 10.1145/1953163.1953241, 2011.
21. A. Repenning, D. Webb, A Ioannidou , "Scalable Game Design and the Development of a Checklist for Getting Computational Thinking into Public Schools", SIGCSE '10 Proceedings of the 41st ACM technical symposium on Computer science education , Milwaukee, WI USA, 2010, pp265-269, DOI: 10.1145/1734263.1734357.
22. P Phillips., "Computational Thinking: A Problem-Solving Tool for Every Class room", 2007 available at http://www.cs.cmu.edu/~CompThink/resources/ct_pat_phillips.ppt#257,1,Computational Thinking. Accessed December 07, 2014.
23. A Settle and L Perkovic, "Computational Thinking across the Curriculum: A Conceptual Framework", 2010. Technical Reports. Paper 13,.http://via.library.depaul.edu/tr/13, Accessed December 07, 2014.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

24. www.cs.cmu.edu, Accessed November 22, 2014.
25. http://www.cs.hmc.edu/~hadas/IntroCourses.html, Accessed December 08, 2014.
26. http://www.jsums.edu/computational-thinking/, Accessed December 08, 2014.
27. http://www.clear.rice.edu/comp140/, Accessed January 12, 2015.
28. http://compthink.cs.depaul.edu/compThink.html, Accessed December 08, 2014.
29. http://web.mst.edu/~vojtat/teaching/cottrell.pdf. Accessed November 10, 2015.
30. Harvey Gould, Jan Tobochnik, "Integrating Computation into the Physics Curriculum",Computational Science — ICCS 2001 Volume 2073 of the series Lecture Notes in Computer Science pp 1031-1040.
31. http://kdeg.scss.tcd.ie/1st-international-workshop-computational-history.
32. Hong Qin, "Teaching computational thinking through bioinformatics to biology students", In Proceedings of the 40th ACM technical symposium on Computer science education (SIGCSE '09). ACM, New York, NY, USA, 188-191. DOI=http://dx.doi.org/10.1145/1508865.1508932.
33. Swati V. Chande, "Computational Sustainability: An Emerging Research and Academic Discipline", International Journal of Advances in Computer Science and Technology, 3(8), August 2014, 421-425.
34. Alex Ruthmann, Jesse M. Heines, Gena R. Greher, Paul Laidler, and Charles Saulters, II. 2010. Teaching computational thinking through musical live coding in scratch. In Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10). ACM, New York, NY, USA, 351-355. DOI=http://dx.doi.org/10.1145/1734263.1734384.

BIOGRAPHY

Dr. Swati V. Chande has over 24 years of experience in teaching, industry, and research. She also has extensive corporate and technical training experience. She has her M.S. in Software Systems of Birla Institute of Technology and Science, Pilani, Doctorate from Banasthali University, and also is a M.Sc. in Mathematics. Her research interests include database management, genetic algorithms, and software engineering.