

### About The Book

**C**omputational thinking is a fundamental skill for everyone, not just computer scientists. Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information processing agent. *Teaching Computational Thinking* introduces the fundamental principles of communicating computing to learners across all levels. The book delves into the philosophical and psychological foundations of computer science as a school subject as well as specific teaching methods, curriculum, tools, and research approaches in computing education. This book is intended as a guide and teaching companion for pre-service and in-service computer science teachers.

### About The Author



Joshua Abah ABAH is a researcher in the Department of Mathematics Education, Federal University of Agriculture, Makurdi, Nigeria. His research interests are Technology of Mathematics Education, Ethnomathematics, Statistical Computing, and Instructional Design. He is happily married to Christy and the marriage is blessed with two children, Prime and Hephzibah.



ISBN:

TEACHING COMPUTATIONAL THINKING

Joshua Abah ABAH



# TEACHING COMPUTATIONAL THINKING

Joshua Abah ABAH

# TEACHING COMPUTATIONAL THINKING

Joshua Abah ABAH



**Network for Grassroots Science and Mathematics Education**

**(The VillageMath Network)**

**<https://ngsme.villagemath.net>**



# **TEACHING COMPUTATIONAL THINKING**

**Joshua Abah ABAH**

Copyright © 2020 Joshua Abah ABAH  
Network for Grassroots Science and Mathematics Education  
(The VillageMath Network)  
Department of Mathematics Education  
Federal University of Agriculture, Makurdi, NIGERIA

# PREFACE

**E**very child in every classroom, every subject teacher in every school, and every person in every community is affected by technology, and the roots of technology are based on the fundamentals of computer science. Beyond becoming educated users of technology, a fundamental understanding of computer science presents students with the possibility of designing and building future technological tools. In addition, the teaching and learning of computer science exposes students to the vital skills of computational thinking.

Computational thinking involves solving problems, designing systems, and understanding human behaviour by drawing on concepts fundamental to computer science. Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. This model of thinking utilizes concepts such as data collection, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization, and simulation. Although computational thinking exists across several subjects and disciplines, the premise of this book is that computer science teachers are better positioned to inculcate the core tenets necessary for developing 21<sup>st</sup> century skill sets. This book introduces computer science teachers (both pre-service and in-service) to the methodology of communicating this style of thought that interprets the world as algorithmically controlled conversions of inputs and outputs.

*Teaching Computational Thinking* takes the reader through an insightful panorama of computing education. The book first x-rays the philosophical foundations of computer science as a science before discussing education in the context of the information age. This is followed by an in-depth consideration of computational thinking and an overview of programming. In a unique presentation, the book delved into an exhaustive exposition on programming paradigms and their influence on the selection of first programming language. As a computer science teacher preparation text, the book dwelt heavily on teaching methods, curriculum, and tools for teaching computer science. A short chapter on research in computer education is intended to expose readers to the existing opportunities for scholarly investigation by computer education experts and students.

It is important to emphasize from the onset that computational thinking is not programming. It is a way that humans, not computers, think. So, *Teaching Computational Thinking* is not a computer programming textbook, though much reference to programming is made throughout the book. Additionally, the book does not endorse any programming language as being superior to others. Comparisons of programming paradigms and languages seen in this book are only from an educational point of view. Having established these vital clarifications, it is suggested that readers use the materials in this text as a guide for teaching computer science to learners across all levels.

**Joshua Abah ABAH**

2020.

# TABLE OF CONTENT

Title Page	-	-	-	-	-	-	i
Copyright Page	-	-	-	-	-	-	iii
Preface	-	-	-	-	-	-	iv
<b>CHAPTER ONE</b>							
The Science of Computer Science	-	-	-	-	-	-	1
Introduction	-	-	-	-	-	-	1
Sources of Knowledge	-	-	-	-	-	-	2
The Tenets of Science	-	-	-	-	-	-	8
Computer Science as a Discipline	-	-	-	-	-	-	18
Why Computer Science is a Science Field	-	-	-	-	-	-	23
<b>CHAPTER TWO</b>							
Education in the Information Age	-	-	-	-	-	-	33
Digital Revolution in Education	-	-	-	-	-	-	33
Impact of Computer Science	-	-	-	-	-	-	37
Computer Science, Information Technology, and Fluency-	-	-	-	-	-	-	40
<b>CHAPTER THREE</b>							
Computation Thinking	-	-	-	-	-	-	42
Definitions and Overview	-	-	-	-	-	-	42
Characteristics of Computational Thinking	-	-	-	-	-	-	50
Computational Thinking Concepts, Capabilities, Disposition and Classroom Culture	-	-	-	-	-	-	51
The Key Techniques of Computational Thinking	-	-	-	-	-	-	64
Assessing Computational Thinking	-	-	-	-	-	-	65
Benefits of Computational Thinking	-	-	-	-	-	-	72
<b>CHAPTER FOUR</b>							
Overview of Programming	-	-	-	-	-	-	74

Introduction - - - - -	74
What is Programming?	76
Programming Languages - - - -	77
The Role of Translators - - - -	87
Stages of Translation - - - -	90
Programming in the Context of Computational Thinking -	92

## CHAPTER FIVE

Common First Programming Languages (FPLS) -	96
Introduction - - - - -	96
Goals and Objectives of a First Programming Course -	99
Main Obstacles in Learning a Specific Programming Language - - - - -	101
Choosing Appropriate First Programming Languages -	105
The Parker Selection Criteria - - -	107
The Farooq Evaluation Framework - - -	118

## CHAPTER SIX

Programming Paradigms in the Context of the First Programming Language - - - -	135
Introduction - - - - -	135
Imperative Paradigm - - - - -	140
Object-Oriented Paradigm - - - -	142
Functional Paradigm - - - - -	147
Logic Paradigm - - - - -	149

## CHAPTER SEVEN

Teaching Methods in Computer Science - -	152
Clarifying Existing Terminologies - - -	152
Approaches - - - - -	152
Methods - - - - -	159
Strategies - - - - -	161
The Teaching Process - - - - -	162



Some Methods for Teaching Computer Science	-	166
Active Learning Techniques	- - - -	212
Learners Alternatives Conceptions	- - -	219
Technological Pedagogical Content Knowledge (TPACK)-		223

## **CHAPTER EIGHT**

Curriculum Issues in Computer Science	- -	228
What is Curriculum?	- - - -	228
Curricular Goals in Computer Science	- -	229
The Nerdc Computer Science Curriculum	- -	231
Sequencing, Spacing and Spiralling	- - -	238
Understading Computing before Programming	-	240
Computational Thinking in the Computer Science Curriculum	- - - - -	242

## **CHAPTER NINE**

Digital Tools for Teaching Computational Thinking -		245
Low Floor, High Ceiling	- - - -	245
Alice	- - - - -	247
Scratch	- - - - -	251

## **CHAPTER TEN**

Research Opportunities in Computer Science Education -		260
Introduction	- - - - -	260
Rationale for Research in Computer Science Education -		265
Computer Science Education Research Categories	-	268
Computer Science Teachers as Researchers	- -	270

<b>REFERENCES</b>	- - - - -	<b>279</b>
-------------------	-----------	------------

# HOW TO OBTAIN THIS BOOK

A COPY OF THIS BOOK COST ₦3000.00 ONLY. This price does not include handling and shipping charges. Shipping can be arranged during book request.

You can request a copy of this book through the following channels:

Email: [editor.villagemath@gmail.com](mailto:editor.villagemath@gmail.com) ; [info@villagemath.net](mailto:info@villagemath.net) ; [ngsme@villagemath.net](mailto:ngsme@villagemath.net)

SMS: +2348073695770 ; +2348179456590

WhatsApp: +2347039240534

Telegram: +2347039240534

For further Information please visit <https://villagemath.net>

## REFERENCES

- Abah, J. A. (2017). Original higher education experience of graduating students of mathematics education in Nigeria: An autoethnographic approach. *Education Journal*, 6(6), 177-187. DOI: 10.11648/j.edu.20170606.13
- Abah, J. A. (2017). Viewing basic math through the lens of history: Undergraduates' reflective learning in a history-augmented mathematics classroom. *Waikato Journal of Education*, 22(4), 33-48. DOI:10.15663/wje.v22i4.557
- Abah, J. A. (2020). An Appeal in the Case involving Conventional Teaching: Emphasizing the Transformation to Enhanced Conventional Teaching in Mathematics Education. *VillageMath Educational Review (VER)*, 1(1), 1-10. DOI: 10.5281/zenodo.3860320.
- Abah, J. A. (2020). *Research practice and ethics in mathematics education*. Makurdi: Network for Grassroots Science and Mathematics Education (The VillageMath Network).
- Abah, J. A., Anyagh, P. I. & Age, T. J. (2017). A flipped applied mathematics classroom: Nigerian university students' experience and perceptions. *ABACUS: The Journal of the Mathematical Association of Nigeria*, 42(1), 78-87.
- Abakpa, B. O., Agbo-Egwu, A. O. & Abah, J. A. (2017). Emphasizing phenomenology as a research paradigm for interpreting growth and development in mathematics education. *ABACUS: The Journal of the Mathematical Association of Nigeria*, 42(1), 391-405. DOI 10.17605/OSF.IO/SGBZT.
- Addair, T. (2014). Why is computer science a science? Retrieved from <https://www.quora.com/Why-is-computer-science-a-science>
- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.
- Ali, A. & Smith, D. (2014). Teaching an introductory programming language in a general education course. *Journal of Information Technology Education: Innovations in Practice*, 13, 57-67.
- Altstadt, D. (2016). *Flipping the classroom: Austin community college's blended, interactive approach to strengthening student progression*. Virginia: The Aspen Institute. pp 1-5.
- Anigbogu, S. O. (2003). *Introduction to computer science and programming languages*. Awka: Christon International Company Ltd.
- Anyor, J. W. & Abah, J. A. (2014). Mathematics curriculum change and assessment models: The quest for an integrated approach. *Benue Journal of Mathematics and Mathematics Education*, 1(3), 11-19. A Publication of the Benue Chapter of the Mathematical Association of Nigeria. [Available from: <https://files.osf.io/v1/resources/egph4/providers/osfstorage/57ea8b72594d9001f491a1fa> ]
- Ary, D., Jacobs, L. C. & Sorensen, C. K. (2010). *Introduction to research in education*. Eight Edition.

Belmont, CA: Wadsworth.

- Atwood, J. (2007). Classic computer science puzzles. Retrieved from <https://blog.codinghorror.com/classic-computer-science-puzzles/amp/>
- Baez, G., Kechiche, B. & Boguszewska, S. (2010). *The impact of mobile services in Nigeria: How mobile technologies are transforming economic and social activities*. Cambridge, MA: Pyramid Research. 3-145.
- Barr, V. & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54.
- Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C. & Marshall, K. S. (2011). Recognizing computational thinking patterns. In *Proceedings of the 42<sup>nd</sup> ACM Technical Symposium on Computer Science Education*, pp. 245-250.
- Bell, T., Alexander, J., Freeman, I. & Grimley, M. (2008). Computer science without computers: New outreach methods from old tricks. In S. Mann & M. Lopez (Eds.), *21<sup>st</sup> Annual Conference of the National Advisory Committee on Computing Qualifications (NACCCQ 2008)*, Auckland, New Zealand, pp. 127-133.
- Bennett, V. E., Koh, K. & Repenning, A. (2013). Computing creativity: Divergence in computational thinking. *Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13)*.
- Bradford, A. (2017). What is science? Retrieved from <https://www.livescience.com/20896-science-scientific-method.html>
- Brennan, K. & Rosnick, M. (2012). Using artifact-based interviews to study the development of computational thinking in interactive media design. Paper presented at the annual American Educational Research Association Meeting (AERA '12). Vancouver, BC, Canada.
- Burke, Q., Bailey, C. S. & Ruiz, P. (2019). Assessing computational thinking. Retrieved from <https://circlcenter.org/assessing-computational-thinking/>
- Business Dictionary (2019). Computer science. Retrieved from <http://www.businessdictionary.com/definition/computer-science.html>
- BusinessDictionary (2019). Computer programming. Retrieved from <https://www.businessdictionary.com/definition/computer-programming>
- Butler, Z., Bezakova, I. & Fluet, K. (2017). Pencil puzzles for introductory computer science: An experience and gender-neutral context. *ACM SIGCSE '17*, March 8-11, 2017, Seattle, WA, USA. DOI: <http://dx.doi.org/10.1145/3017680-3017765>
- Cassady, L. (2014). *What are the primary purposes of education?* Retrieved on 18<sup>th</sup> March, 2014 from [http://www.ehow.com/m/info\\_8192984\\_primary\\_purpose\\_education.html](http://www.ehow.com/m/info_8192984_primary_purpose_education.html)
- Cerf, V. G. (2012). Where is the science in computer science? *Communications of the ACM*,

55(10), 5. DOI: 10.1145/2347736.2347737

- Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, 17, 93-100.
- Chaparro, E. A., Yuksel, A., Romero, P. & Bryant, S. (2005). Factors affecting the perceived effectiveness of pair programming in higher education. Proceedings of the 17<sup>th</sup> Workshop of the Psychology of Programming Interest Group, pp. 5-18.
- Charles-Ogan, G. & Williams, C. (2015). Flipped classroom versus a conventional classroom in the learning of mathematics. *British Journal of Education*, 3(6), 71-77
- Chow, K. C. K., Chu, S. K. W., Tavares, N. & Lee, C. W. Y. (2015). Teachers as researchers: A discovery of their emerging role and impact through a school-university collaborative research. *Brock Educational Journal*, 24(2), 20-39.
- Cochran-Smith, M. & Lytle, S. (1999). The teacher research movement: A decade later. *Educational Researcher*, 28(7), 15-25.
- Code Club (2018). *Simple coding for total beginners: Book of Scratch*. Volume 1. Cambridge: Raspberry Pi Press
- Colmenares, A. (2014). *Technology for flipping the classroom: Tips and trends*. Chicago, IL: Association of College and Research Libraries. pp 1-4
- Condon, A., Frost, D., Guzdial, M., Sutner, K. & Williams, L. (2003). Forward to the ACM model curriculum. New York, NY: CSTA
- Condon, A., Frost, D., Guzdial, M., Sutner, K. & Williams, L. (2003). *Forward to the ACM model curriculum*. New York, NY: CSTA
- Cooper, S., Forbes, J., Fox, A., Hambruch, S., Ko, A. & Simon, B. (2016). The importance of computing education research. *A Whitepaper for Computing Community Consortium (CCC)*, pp. 1-11.
- Craig, D. J. (2009). *Defining a 21<sup>st</sup> century education*. Alexandria, VA: The Centre for Public Education, 1-79.
- Curzon, P. (2016). *Computational thinking: Cut hive logic puzzles*. London: Queen Mary University of London.
- D'Ostualdo, E. (2017). Why computer science is a science. Retrieved from <https://www.emmanueldosualdo.com/blog/2017/why-computer-science-is-a-science.html>
- Dagiene, V. & Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. In *Proceedings of International Conference on Informatics in Secondary Schools - Evolution and Perspectives*, pp. 19-30.
- Dann, W., Cooper, S. & Pausch, R. (2006). *Learning to program with Alice*. New Jersey: Pearson.

- Dann, W., Slater, D., Paoletti, L., Casgrove, D., Culyba, D. & Tang, P. (2014). Alice 3: How to guide. Retrieved from <https://www.alice.org>
- Denning, P. J. (2007). Computer science: Is it really a science and what's science about? A published interview with Peter J. Denning. Retrieved from <https://www.wired.com/2007/06/computer-science/>
- Denning, P. J. (2009). The profession of IT: Beyond computational thinking. *Communications of the ACM*, 52(6), 28-30. DOI: 10.1145/1516045.1516054
- Dictionary.Com (2019). Computer science. Retrieved from <https://www.dictionary.com/browse/computer-science>
- Dorling, M. (2016). Computational thinking rubric: Learning behaviours, dispositions and perspectives. Retrieved from <https://community.computingatschool.org.uk>
- Duffany, J. L. (2014). Choice of language for an introduction to programming course. A paper presented at the 12<sup>th</sup> Latin American and Caribbean Conference for Engineering and Technology (LACCET' 2014): *Excellence in Engineering to Enhance a Country's Productivity*. July 22-24, 2014, Guayaquil, Ecuador. pp. 1-9.
- Duncan, C. (2018). Reported development of computational thinking through computer science and programming and its benefits for primary school students. In *SIGCSE '18. Proceedings of the 49<sup>th</sup> ACM Technical Symposium on Computer Science Education*. Baltimore, Maryland, USA.
- Durak, H. Y. & Saritepeci, M. (2017). Analysis of the relation between computational thinking skills and various variables with the structured equation model. *Computers & Education*, 116, 191-202.
- Edutopia (2007). *What is successful technology integration?* Retrieved on 15<sup>th</sup> November, 2016 from <http://www.edutopia.org/technology-integration-guide-description>
- Emaikwu, S. O. (2015). *Fundamentals of research methods and statistics*. Makurdi: Selfers Academic Press Limited.
- Falkner, N., Sooriamurthi, R. & Michalewicz, Z. (2010). Puzzle-based learning for engineering and computer science. *IEEE Computer*, 43(4), 20-28.
- Farooq, M. S., Khan, S. A., Ahmad, F., Islam, S. & Abid, A. (2014). An evaluation framework and comparative analysis of the widely used first programming languages. *PLoS ONE*, 9(2), e88941.
- Fee, S. B. & Holland-Minkley, A. M. (2010). Teaching computer science through problems, not solutions. *Computer Science Education*, 20(2).
- Fernandez-Villaverde, J., Guerron, P. & Valencia, O. Z. (2019). Programming paradigms: Lectures on high-performance computing for economists. VII. Retrieved from [https://www.sas.upenn.edu/~jesusfv/Uncertainty\\_Shocks\\_Business\\_Cycle.pdf](https://www.sas.upenn.edu/~jesusfv/Uncertainty_Shocks_Business_Cycle.pdf)

- Flipped Learning Framework (2014). *Definition of flipped learning*. Retrieved from <http://www.flippedlearning.org/definition>
- Fulton, K. (2012). Upside down and inside out: Flip your classroom to improve student learning. *Learning & Leading with Technology*, 39(8), 12-17.
- GeeksforGeeks (2020). Find last two remaining elements after removing median of any 3 consecutive elements repeatedly. Retrieved from <https://www.geeksforgeeks.com/find-last-two-remaining-elements-after-removing-median-of-any-3-consecutive-elements-repeatedly>
- Gehring, E. F. (2007). Active and collaborative learning strategies for teaching computer science. Annual Conference and Exposition, American Society for Engineering Education, Honolulu, HI, June 24-27, 2007.
- Gemmell, M. (2002). Introduction to programming: What programming actually is. Retrieved from <http://www.scotlandsoftware.com/article/programming>
- Gill, A. K. & Kusum (2017). Teaching approaches, methods and strategy. *Scholarly Research Journal for Interdisciplinary Studies*, 4(36), 6692-6697. DOI: 10.21922/srjis.v4i36.10014
- Gray, J. & Campbell-Evans, G. (2002). Beginning teachers as teacher-researchers. *Australian Journal of Teacher Education*, 27(1). DOI: 10.14221/ajte.2002v27n1.4
- Griffiths, R. (2004). Knowledge production and the research-teaching nexus: The case of the built environment disciplines. *Studies in Higher Education*, 29(6), 709-726.
- Grover, S. & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. DOI: 10.3102/0013189X12463051
- Grover, S., Basu, S. & Schank, P. (2018). What we can learn about student learning from open-ended programming in middle school computer science. In *49<sup>th</sup> ACM Technical Symposium on Computer Science Education (SIGCSE '18)*, Baltimore, MD, USA.
- Guzdial, M. (2008). Education: Paving the way for computational thinking. *Communications of the ACM*, 51(8), 25-27. DOI: 10.1145/1378704.1378713
- Haridi, S. & Franzen, N. (2008). Tutorial of Oz. Retrieved from <http://mozart2.org/mozart-v1/doc-1.4.0/tutorial/index.html>
- Hazzan, O., Lapidot, T. & Ragonis, N. (2011). *Guide to teaching computer science*. London: Springer
- Healey, M. (2005). Linking research and teaching: Disciplinary spaces. In R. Barnett (Ed.), *Reshaping the University: New Relationships between Research, Scholarship and Teaching*. pp 30-42. Maidenland: McGraw-Hill/Open University Press
- Heo, H. J. & Choi, M. R. (2014). Flipped learning in the middle school math class. *Advanced Science and Technology Letters*, 71, 94-97.

- Hu, X. (2018). Oz: A language being way too academic? Retrieved from <https://medium.com/elp-2018/oz-a-language-being-way-too-academic-bbe6a5c2b911>
- Iji, C. O., Abah, J. A. & Uka, N. K. (2013). Attaining the Millenium Development Goals (MDGs) through Effective Mathematics Education. *Proceedings of the 54<sup>th</sup> Annual Conference of Science Teachers Association of Nigeria*, 99-107.
- Ingram, D., Wiley, B., Miller, C. & Wyberg, T. (2014). *A study of the flipped math classroom in the elementary grades*. Saint Paul, MN: University of Minnesota, college of Education and Human Development, Center for Applied Ressearch and Educational Improvement. pp 1-25
- Innovation Unit (2014). *21<sup>st</sup> century learning*. Retrieved on 18<sup>th</sup> March, 2014 from <http://www.innovationunit.org/knowledge/our-ideas/21st-century-learning>
- International Working Group on Computational Thinking - IWGCT (2010). *Computational thinking for youth*. Newton, MA: Education Development Center Inc.
- Italiano, E. (2014). *Community, contemplation, and computers: The role of technology in education*. Retrieved on 18<sup>th</sup> March, 2014 from <http://www.thepublicdiscourse.com/2014/02/11789/>
- Kadry, S. & El Hami, A. (2014). Flipped classroom model in Calculus II. *Education*, 4(4), 103-107.
- Kahanwal, B. (2013). Abstraction level taxonomy of programming language frameworks. *International Journal of Programming Languages and Application (IJPLA)*, 3(4), 1-12. DOI: 10.5121/ijpla.2013.3401
- Ketteridge, S. & Shiach, M. (2009). Supervising research students. In H. Fry, S. Ketteridge & S. Marshal (Eds.), *A Handbook for Teaching and Learning in Higher Education: Enhancing Academic Practice*, 3<sup>rd</sup> Edition. pp 166-185. New York: Routledge.
- Kharitonova, Y. (2008). Overview of issues in computer science and recommendation for teaching computer science classes: An innovative approach to introductory computer science courses. *Hohonu*, 6, 10-15.
- Kirkwood, A. & Price, L. (2014). Technology-enhanced learning and teaching in higher education: What is “enhanced” and how do we know? A critical literature review. *Learning, Media and Technology*, 39(1), 6-36.
- Koehler, M. J. & Mishra, P. (2008). Introducing TPACK. In AACTE Committee on Innovation & Technology (Eds.), *Handbook of Technological Pedagogical Content Knowledge for Educators*, pp. 3-29. New York, NY: Routledge
- Koehler, M. J., Mishra, P., Kereluik, K., Shin, T. S. & Graham, C. R. (2014). The technological pedagogical content knowledge framework. In J. M. Spencer et al. (Eds.), *Handbook of Research on Educational Communications and Technology*, pp. 101-111. New York, NY: Springer Science + Business Media. DOI: 10.1007/978-1-4614-3185-5\_9
- Koh, K. H., Basawapatna, A., Nickerson, H. & Repenning, A. (2010). Real time assessment of



computational thinking. *Proceedings of IEEE '14, Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. Melbourne, Australia.

- Korkmaz, O, Cakir, R. & Ozden, M. Y. (2017). A validity and reliability study of computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558-569.
- Koulouri, T., Lauria, S. & MacRedie, R. D. (2014). Teaching introductory programming: A quantitative evaluation of different approaches. *ACM Transactions on Computing Education (TOCE)*, 14(4), 1-28.
- Kramer, J. (2007). Is abstraction the key to computing? *Communications of the ACM*, 50(4), 37-42.
- Kruglyk, V. & Lvov, M. (2012). Choosing the first educational programming language. *ICTERI 2012*. pp. 188-198.
- LanCastle, S., Barlow, C. & Davison, L. (2016). A flipped classroom in engineering education - student perception and effect on learning. *ICICTE 2016 Proceedings*. pp 323-332.
- Larsen, J. (2015). Adult students' experiences of flipped mathematics classroom. *Adults Learning Mathematics: An International Journal*, 10(1), 50-67.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ..., Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2, 32-37.
- Levitin, A. & Levitin, M. (2011). *Algorithmic puzzles*. Oxford: Oxford University Press
- Lewis, C. (2009). Human-centred computing and representation: A framework. CHI 2009, April 4-9, 2009. Boston, MA: ACM
- Lingard, B., Hayes, D. & Mills, A. (2003). Teachers and productive pedagogies: Contextualizing, conceptualizing, utilizing. *Pedagogy, Culture & Society*, 11(3), 73-86.
- Lokesh, U. (2013). *Technology and its role in 21<sup>st</sup> century education*. Retrieved on 18<sup>th</sup> March, 2014 from <http://www.edtechreview.in/trends-insights/insights/277-role-of-technology-in-21st-century>
- Lu, J. J. & Fletcher, G. H. (2009). Thinking about computational thinking. SIGCSE' 09, March 3-7, 2009, Chattanooga, Tennessee, USA. pp. 1-5.
- Lysaker, J. & Thompson, B. (2013). Teacher research as a practical tool for learning to teach. *Language Arts*, 90(3), 181-191.
- Madushanka, L. (2018). Solving river-crossing puzzles using python. Retrieved from <https://www.google.com/amp/s/ahirumadushankablog.wordpress.com/2017/10/27/Solving-river-crossing-puzzles-using-python/amp/>
- Marji, M. (2014). *Learn to program with Scratch: A visual introduction to programming with games, art, science, and math*. San Francisco, CA: No Starch Press

- Marshall, S. (2009). Supervising projects and dissertations. In H. Fry, S. Ketteridge & S. Marshall (Eds.), *A Handbook for Teaching and Learning in Higher Education: Enhancing Academic Practice*, 3<sup>rd</sup> Edition. pp 150-165. New York: Routledge.
- Marshall, K. S. (2011). Was that CT? Assessing computational thinking through video-based prompts. A paper presented at the 2011 Annual Meeting of the American Educational Research Association, New Orleans, Louisiana, USA.
- McLaughlin, C. (2004). Practitioner research and enquiry. In C. McLaughlin, K. Black-Hawkins & D. McIntyre (Eds.), *Researching Teachers, Researching Schools, Research Networks: A Review of the Literature*. pp 4-20. Cambridge: University of Cambridge.
- McLaughlin, J. E., Roth, M. T., Glatt, D. M., Gharkholonarche, N., Davidson, C. A., Griffin, L. M., Esserman, D. A. & Mumper, R. J. (2014). The flipped classroom: A course redesign to foster learning and engagement in a health professions school. *Academic Medicine*, 89(2), 236-243.
- Merrick, K. E. (2010). An empirical evaluation of puzzle-based learning as an interest approach for teaching introductory computer science. *IEEE Transactions on Education*, 53(4), 677-680.
- Miller, L. A. (1981). Natural language programming: Styles, strategies, and contrasts. *IBM Systems Journal*, 29(2), 184-215.
- Mohorovicic, S. Strcic, V. (2011). An overview of computer programming teaching methods. Central European Conference on Information and Intelligent Systems.
- Muir, T. & Chick, H. (2014). Flipping the classroom: A case study of a mathematics methods class. In J. Anderson, M. Cavanagh & A. Prescott (Eds.), *Curriculum in Focus: Research Guided Practice*, Proceedings of the 37th Annual Conference of the Mathematics Education Research Group of Australasia. Sydney: MERGA. pp 485-492
- Murphy, E., Crick, T. & Davenport, J. H. (2017). An analysis of introductory programming courses at UK universities. *The Art, Science, and Engineering of Programming*, 1(2), Article 18, 23 pages.
- National Research Council - NRC (2010). *Report of a workshop on the scope and nature of computational thinking*. Washington, DC: The National Academic Press
- National Universities Commission - NUC (2007). *Benchmark minimum academic standards for undergraduates programmes: Education*. Abuja: NUC. Pp 1-353.
- Osigbeme, M., Eze, M., Imouokhome, F. & Omenka, O. (2015). Achieving modified Bloom's taxonomy with the Zolla class management and learning software. *African Journal of Computing & ICT*, 8(3), 9-16.
- Pane, J. F., Ratanamahatana, C. & Meyers, B. A. (2001). Studying the language and structure in non-programmers' solutions to programming problems. *International Journal of Human-Computer Studies*, 54, 237-264.
- Parker, K. R., Ottaway, T. A. & Chao, J. T. (2006). Criteria for the selection of a programming

- language for introductory courses. *International Journal of Knowledge and Learning*, 2(1/2), 119-139.
- Partnership for 21<sup>st</sup> Century Skills (2002). *Learning for the 21<sup>st</sup> century: A report and mile guide for 21<sup>st</sup> century skills*. Washington: Partnership for 21<sup>st</sup> Century Skills, 1-5.
- Perkovic, L., Settle, A., Hwang, S. & Jones, J. (2010). A framework for computational thinking across the curriculum. *ITiCSE'10*, June 26-30, 2010, Bilkent, Ankara, Turkey. pp. 123-127.
- Powers, K. D. & Powers, D. T. (1999). Making sense of teaching methods in computing education. 29<sup>th</sup> ASEE/IEEE Frontiers in Education Conference, November 10-13, 1999, San Juan, Puerto Rico, 11b3, pp. 30-35.
- Pressman, R. S. (2005). *Software engineering: A practitioner's approach*. 6<sup>th</sup> Edition. New York, NY: McGraw Hill Companies, Inc.
- Puarungroj, W. (2015). Inverting a computer programming class with the flipped classroom. *Proceedings of the International Conference on eLearning for Knowledge-Based Society* held 11-12 December, 2015, in Thailand. pp 40.1-40.7
- Raja, R. S. (2002). *Education for the twenty-first century: Asia-Pacific Perspective*. Bangkok: UNESCO, 1-111.
- Reci, E. & Bollin, A. (2017). Managing the quality of teaching in computer science education. CSERC '17, November 14, 2017, Helsinki, Finland. <https://doi.org/10.1145/3162087.3162097>
- Repenning, A., Webb, D. & Ioannidou, A. (2010). Scalable game design and the development of checklist for getting computational thinking into public schools. In *Proceedings of the 41<sup>st</sup> ACM Technical Symposium on Computer Science Education (SIGCSE '10)*, 265-269. New York, NY: ACM Press.
- Resnick, M., Maloney, J., Hernandez, A. M., Rusk, N., Eastmond, E., ..., Kefai, Y. (2009). Scratch: programming for everyone. *Communications of the ACM*, 52(11), 60-67.
- Roman-Gonzalez, M., Moreno-Leon, J. & Robles, G. (2017). Complementary tools for computational thinking assessment. *Proceedings of International Conference on Computational Thinking Education 2017*, Hong Kong, China.
- Roman-Gonzalez, M., Perez-Gonzalez, J. C. & Jimenez-Fernandez, C. (2017). Which cognitive abilities underlie computational? Criterion validity of the computational thinking test. *Computers in Human Behavior*, 72, 678-691.
- Ross, K., Dennis, B., Zhao, P. & Li, P. (2017). Exploring graduate students' understanding of research: Links between identity and research conceptions. *International Journal of Teaching and Learning in Higher Education*, 29(1), 73-86.
- Rust, F. O. (2009). Teacher research and the problems of practice. *Teachers College Record*, 111(8), 1882-1893.

- Salleh, H. (2004). Why should teachers do research? *SingTeach*, 46, 1-3.
- ScienceBuddies (2019). Steps of the scientific method. Retrieved from <https://www.sciencebuddies.org/science-fair-projects/science-fair/steps-of-the-scientific-method>
- Shafique, M. & Irwin-Robinson, H. (2015). A study on the effectiveness of flipped teaching in college math classroom. *International Journal of Education and Information Technology*, 1(2), 29-33. Retrieved on 15th July, 2016, from <http://www.publicscienceframework.org/journal/ijeit>
- Sherman, S. J., Shehane, R. F. & Todd, D. W. (2018). Quantitative model for choosing programming language for online instruction. *Journal of Instructional Pedagogies*, 20.
- Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, 15(2), 4-14.
- Slavin, T. (2014). Computer science unplugged. Retrieved from <https://kidscodecs.com/computer-science-unplugged>
- Smith, G. & Fidge, C. (2008). On the efficacy of prerecorded lectures for teaching introductory programming. *Proceedings of the Tenth Conference on Australasian Computing Education*, 78, 129-136.
- Sung, K., Hillyard, C., Angotti, R. L., Panitz, M. W., Goldstein, D. S. & Nordlinger, J. (2010). Game-themed programming assignment modules: A pathway for gradual integration of gaming context into existing introductory courses. *IEEE Transactions on Education*, Vol. PP, Issue 99, 1-12.
- Taylor, E., Breed, M., Hauman, I. & Homann, A. (2013). Choosing learning methods suitable for teaching and learning in computer science. *IADIS International Conference on e-Learning 2013*. pp. 74-82.
- Technopedia (2019). Computer science definition - what does computer science mean? Retrieved from <https://www.technopedia.com/definition/592/computer-science>
- Technopedia (2019). Programming. Retrieved from <https://www.technopedia.com/definition/13128/programming>
- TechTerms (2019). Computer science definition. Retrieved from [https://techterms.com/definition/computer\\_science](https://techterms.com/definition/computer_science)
- Triantafyllou, E., Timcenko, O. & Kofoed, L. B. (2015). *Student behaviours and perceptions in a flipped classroom: A case in Undergraduate Mathematics*. A paper presented at the 43rd Annual SEFI Conference June 20-July 2, 2015 at Orleans, France.
- Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C. & Verno, A. (2003). *A model curriculum for K-12 computer science: Final report of the ACM K-12 task force curriculum committee*. New York, NY: CSTA

- Uesugi, S. (2016). My favorite puzzle to introduce computer science to teenagers. Retrieved from <https://medium.com/@chibicode/my-favorite-computer-science-puzzle-to-introduce-algorithms-to-tenagers-a7b8159a5d63>
- Van Roy, P. & Haridi, S. (2003). *Concepts, techniques, and models of computer programming*. Louvain-la-Neuve: Universite Catholique de Louvain.
- Van Roy, P. (2009). Programming paradigms for dummies: What every programmer should know. Retrieved from <http://hiperc.buffalostate.edu/courses/ACM612-F15/uploads/ACM612/VanRoy-Programming.pdf>
- Vasquez, V. E. L. (2017). *Teachers as researchers: Advantages, disadvantages and challenges for teachers intending to engage in research activities*. Retrieved on 21<sup>st</sup> April, 2018 from <https://www.academia.edu/719736>
- Vujosevic-Janicic, M. & Totic, D. (2008). The role of programming paradigms in the first programming courses. *The Teaching of Mathematics*, 11(2), 63-83.
- Weese, J. L. & Feldhausen, R. (2017). STEM outreach: Assessing computational thinking and problem solving. Annual Conference and Exposition of the American Society for Engineering Education (ASEE '17): *Curricular Issues in Computing*.
- Weintrop, D. & Wilensky, U. (2015). Using commutative assessments to compare conceptual understanding of blocks-based and text-based programs. *ICER*, 15, 101-110.
- Werner, L, Denner, J., Campe, S. & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. *Proceedings of the 43<sup>rd</sup> ACM Technical Symposium on Computer Science Education (SIGCSE'12)*, 215-220. New York, NY: ACM
- Wikipedia (2019). Computer science. Retrieved from [https://en.m.wikipedia.org/wiki/computer\\_science](https://en.m.wikipedia.org/wiki/computer_science)
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society*, 366, 3717-3725. DOI: 10.1098/rsta.2008.0118
- Wing, J. M. (2011). Research notebook: Computational thinking - what and why? *The Link*.
- Wirth, M. (2018). Solving the “River Problem” puzzle using mutual recursion. Retrieved from <https://craftofcoding.files.wordpress.com/2018/08/rivercrossing.pdf>
- Yoshida, H. (2016). Perceived usefulness of flipped learning on instructional design for elementary and secondary education: With focus on pre-service teachers. *International Journal of Information and Education Technology*, 6(6), 430-434.
- Zacharis, N. (2009). Evaluating the effects of virtual pair programming on students' achievement and satisfaction. *International Journal of Emerging Technologies in Learning*, 4(3), 34-39.

Zacharis, N. Z. (2011). Measuring the effects of virtual pair programming in an introductory programming Java course. *IEEE Transactions on Education*, *54(1)*, 168-170.