COMMUNICATION

LEADCS.org

Computer Science Terminology

QUESTION

What exactly is "computer science?" How does it relate to other terms, like "computational thinking," "programming or coding," "digital literacy," and the like?

TOOL DESCRIPTION

Computer science education terminology can be confusing. This tool provides a list of frequently used terms in the computer science education field and summarizes their most common definitions.

WHY DO YOU NEED TO KNOW THIS?

The computer science education field is filled with many terms that attempt to define and explain the discipline (and closely related disciplines) and its goals. However, educators, scientists, engineers, and others working in computer science related fields do not always define computer-related terms in the same way.

The lack of shared language is a recurring theme. For example, the 2014 ACM report, Rebooting the Pathway to Success says there is "widespread confusion" about what exactly constitutes "academic computer science" (p.17). Further, while computer scientists agree that computer science is a scientific discipline, they do not agree on a "single, unified definition of the discipline" (Gal-Ezer and Stephenson 2010: 62). The "Building an Operating System for Computer Science" (OS4CS) study (Century et al. 2013) found that terminology confusion is a pervasive problem, reporting no commonly accepted computer science definition among teachers, students, school leaders, and professional development providers. The study also showed that educators view computer science in widely varied ways ranging from computer applications (e.g. Microsoft Word), to programming, to logic modeling and problem-solving.

Terminology confusion makes it difficult for K-12 educators and other leaders to clearly describe to others (district administrators,school leaders, other teachers, students, families, etc.) what computer science is and is not, and why it is necessary in the 21st Century.

As part of the work of leading change, having clear definitions about goals and objectives related to computer science educationis essential for productive discussions with different stakeholders. This tool provides some context to the myriad of definitions and terms used in the computer science education field and can be used to drive discussions with stakeholders.

HOW DOES THIS TOOL HELP?

This tool provides a list of often-used terms related to computer science education and summarizes their most common definitions. It also includes a list of "Frequently asked Questions" about computer science terminology. Links to the many sources used to compile these definitions are provided at the end of the tool.

The tool contains definitions for the following terms:

- Computational literacy
- Computational thinking (CT)
- Computational thinking practices
- Computer engineering
- Computer literacy
- Computer science (CS)
- Computing
- Cyberlearning
- Digital literacy
- Educational technology
- Informatics
- Information and communication technology (ICT)
- Information science
- Information systems
- Information technology (IT)
- Programming and coding

TOOL CONTENT

COMPUTER SCIENCE TERMS

Computational Literacy refers to having the knowledge to make use of the power of the computer beyond using typical applications. Computational literacy was a phrase used before computer scientists began talking about **computational thinking**, and the two terms are now often used interchangeably. According to computer science professor Michael Horn, computational literacy is the ability to use computers and computational technologies to solve problems, both in school and in the community. Computational literacy is sometimes confused with the terms **computer literacy** and **digital literacy**, which themselves do not mean the same thing.

Computational Thinking (CT) in computer science refers to a problem solving process used by computer scientists. Jeannette Wing, a computer scientist, describes computational thinking as the thought processes involved in formulating a problem and expressing its solutions in a way that a computer can effectively carry it out (Wing 2014). Computational thinking concepts in K-12 include data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization, and simulation.

It is important to point out that computational thinking "shares elements with various other types of thinking such as algorithmic thinking, engineering thinking, and mathematical thinking (IWG 2010, as cited in Barr and Stephenson 2011). And while computer scientists acknowledge that many subjects teach problem solving skills, they believe that **computer science** often uses a particularly systematic and deep approach to thinking about complex problems, which they call computational thinking. In mathematics, some K-12 instructional materials outline the **computational thinking practices** that students should use to engage with computer science course content. Computational thinking and mathematical thinking are closely related, as both are involved with abstraction and reasoning with recognized simplified models. **Computational literacy** is a phrase that existed before computational thinking, and these two terms are sometimes used interchangeably.

Computational Thinking Practices are the *practices and skills* that computer scientists use, including, for example, abstracting, analyzing problems and [computational] artifacts, communicating, and collaborating. These practices used in computer science are in some ways akin to the Standards for Mathematical Practice, in the Common Core State Standards, or the Science and Engineering Practices in the Next Generation Science Standards.

Computer Engineering is specifically concerned with the design and construction of computers and computer-based systems. This includes computer hardware design and the design of computer-based devices.

Computer Literacy is different from **computational literacy** and **digital literacy** in that computer literacy focuses on the knowledge and ability to use computers (e.g., use of computer components, like the keyboard and mouse; locating files; using special function keys) and existing technologies or applications (e.g., word processing, desktop publishing, spreadsheets; email) efficiently. Essentially, **computer literacy** is about understanding how to work with computers and their related systems.

Computer Science, or CS, is an academic, scientific discipline that includes the study of computers and algorithmic processes (procedures or formulas for solving problems) "including their principles, their hardware and software designs, their applications, and their impact on society" (www.acm.org). The discipline relies on problem solving, design, and logical reasoning, combining human ideas and digital tools in order to create or adapt digital technology, not just use it. While the discipline is often equated with programming, Computer science is *not* just about programming or coding. Programming is the expression of algorithmic thinking. Algorithmic thinking is fundamental to computer science, so computer science classes typically include a programming component. But programming is just one piece of computer science.

Computer science is sometimes referred to as **computing**, which is an inclusive term used for a broad range of computing fields similar to computer science, like **computer engineering, information systems, information technology**, etc.

Computing is a term often used to describe **computer science** and **information technology**-related activities. This broad and inclusive term is also used to indicate a wide range of computer science- related (but separate) fields and areas of study, like **computer science, computer engineering,** software engineering**, information systems, information technology**, and **information science**. These allied fields all involve the creation or adaptation of technology rather than simply using technology. The Association for Computing Machinery (an international educational and scientific computing society) defines computing as any goal-oriented activity requiring, benefiting from, or creating computers.

Cyberlearning refers to the use of networked computing and communications technologies to support learning. As further defined by the National Science Foundation as "learning that is mediated by networked computing and communications technologies" (2008, p. 10), or more simply put, technology that supports learners. Cyberlearning is not the same as computer science education, nor is it the same as **computer literacy, digital literacy,** or **educational technology** because the focus of cyberlearning is on technology specific to supporting learning (C. Hoadley, personal communication, March 26, 2015).

Digital Literacy is different from **computational literacy** and **computer literacy**. Digital Literacy refers to the ability to use digital technology and communication tools or networks to locate, evaluate, use, and create information, as well as interact safely in the digital world. For example, digital literacy includes a person's ability to perform tasks effectively in the digital environment, such as reading and interpreting media, reproducing data and images through digital manipulation, evaluating and applying new knowledge gained from digital environments, and understanding how such technologies impact the world we live in. **Computer literacy** simply refers to the knowledge and ability to use components of the computer (like the keyboard and mouse; locating files; using special function keys) and existing technologies (e.g., word processing, desktop publishing, spreadsheets; email) efficiently.

Educational Technology is a term used in school settings that refers to the integration of technology into teaching, or using computer technology (hardware and software) to enhance learning across academic disciplines. Educational technology sometimes refers to tools for productivity, like smart boards or software for managing a class web site, or to content-specific technology or software.

Informatics refers to the study and application of **information technology** to the arts, science and other professions, and to its use in organizations and society at large. Informatics uses computing to solve big problems, like privacy, security, healthcare, education, poverty, and challenges in our environment. It combines computational, cognitive, and social aspects. People within Informatics say it differs from **computer science** in its specific emphasis on the human use of computing; informatics is the study of *how* people interact with components of **information technology**.

Information and Communication Technology (ICT) refers to technologies that provide access to information through telecommunications. Information and *communication* technology is similar to **information technology**, but focusing on communication technologies, including the Internet, wireless networks, cell phones, and other mediums for communication. When taught in schools, ICT teaches students how to be thoughtful users of computers and computational tools (i.e., tools created by those in **computer science**).

Information Science is the study of information and how people within organizations use information. That is, information science examines the interactions between people and technology, how technology shapes individual lives and social groups, and how the ways people use technology can shape new developments.

Information Systems refers to the integration of **information technology** solutions and business processes to meet the information needs of businesses and other enterprises, enabling them to effectively and efficiently achieve their objectives.

Information Technology (IT) refers to anything related to computing technology that manages and processes information, like networks, hardware, software, or the Internet. Information Technology typically focuses on applying the components of IT to solve business information problems, like network or database administration. Information Technology emphasizes the technology tools and their use whereas **computer science** is concerned with how technological tools are designed and why they work. In many schools, the Career and Technical Education (CTE) program includes IT as a program strand.

Programming and Coding are terms sometimes used interchangeably, although there is currently some debate in the computing community about how to apply these terms appropriately. Traditionally, computer **programming** is seen as the creative act of designing a computer program, or a set of instructions *for a* computer to do a task.

Coding is also involved in the act of programming, but traditionally denotes the physical act of actually writing the code for a program in some programming language. Most computer programmers actually both design and code their own programs. Increasingly the word "coding" has been used as a catch-all for the creative and physical acts of programming, frequently carrying a tone of a less formal approach to creative programming.

FREQUENTLY ASKED QUESTIONS

What is the difference between the terms Computer Science and Computing?

Computer science is a specific academic discipline "that encompasses the study of computers and algorithmic processes including their principles, their hardware and software design, their applications, and their impact on society" (www.acm.org; CSTA 2013, 2011; Tucker 2006). It involves problem solving, design, and logical reasoning. Computing, on the other hand, is a broad and inclusive term. "Computing" is often used to describe computer science and information technology-related activities. The term is also used to indicate a wide range of computer science- related fields and areas of study. The Association for Computing Machinery (an international educational and scientific computing society) further defines computing as any goal-oriented activity requiring, benefiting from, or creating computers.

What does Computational Thinking have to do with Computer Science?

Computational thinking is a term that refers to a problem solving process used by computer scientists. Computer scientists acknowledge that many subjects teach problem solving skills, but they believe that computer science often uses a more systematic and deeper approach to thinking about complex problems, which they call computational thinking. Computational literacy is a phrase that existed before computational thinking, and these two terms are sometimes used interchangeably.

What is the difference between Computational Literacy and Computational Thinking?

"Computational literacy" was an idea discussed before the term computational thinking became widely popular through use by Seymour Papert and Jeannette Wing. Computational literacy was used by Andrea diSessa in *Changing minds: Computers, learning, and literacy* (2000). While computational thinking is the more popular of the two terms, they both refer to a problem solving process, and are often used interchangeably in research and practice.

What are the Computational Thinking Practices? Where can I find more information about these practices?

As described in the AP Computer Science Principles Curriculum Framework, the computational thinking practices include:

- *Connecting computing* (e.g. describing connections between people and computing)
- *Creating computational artifacts* (e.g. selecting appropriate techniques to develop a computational artifact)
- *Abstracting* (e.g. explaining how data, information, or knowledge is represented for computational use)
- *Analyzing problems and artifacts* (e.g. locating and correcting errors)
- *Communicating* (e.g. explaining the meaning of a result in context)
- *Collaborating* (e.g. exchanging knowledge and feedback with a partner or a team member)

What fundamental Computer Science concepts are included in Computational Thinking?

The Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) say Computational Thinking for K-12 includes the following fundamental computer science concepts:

- Data collection
- Data analysis
- Data representation
- Problem decomposition
- Abstraction
- Algorithms and procedures
- Automation
- Parallelization
- Simulation

How is Information Technology different from Computer Science?

Information technology refers to anything related to computing technology that manages and processes information, like networks, hardware, software, or the Internet. Information technology emphasizes the technology tools and their use whereas computer science is concerned with how technological tools are designed and why they work. In many schools, the Career and Technical Education (CTE) program includes IT as a program strand.

Is Educational Technology the same thing as Computer Science?

No, educational technology is a term used in school settings that refers to the integration of technology into teaching, or using computer technology (hardware and software) to enhance learning across academic disciplines whereas computer science is an academic, scientific discipline that involves problem solving, design, and logical reasoning that combines human ideas and digital tools. Computer science examines how computers and computer systems work, how they are constructed and programmed, and the foundations of information and computation.

How is Programming different from Computer Science?

Computer science is an academic, scientific discipline that includes the study of computers and algorithmic processes (procedures or formulas for solving problems) "including their principles, their hardware and software designs, their applications, and their impact on society" (www.acm.org). It combines human ideas and digital tools, and examines how computers and computer systems work, how they are constructed and programmed, and the foundations of information and computation. Programming is the expression of algorithmic thinking. Algorithmic thinking is fundamental to computer science, so computer science classes typically include a programming component. But programming is just one piece of computer science. Programming is seen as the creative act of designing a computer program, or a set of instructions *for a* computer to do a task.

Are Computer Literacy and Computational Literacy the same thing?

No, these terms have different meanings. Computer literacy focuses on the knowledge and ability to use computers (e.g., use of computer components, like the keyboard and mouse; locating files; using special function keys) and existing technologies (e.g., word processing, desktop publishing, spreadsheets; email) efficiently. As defined by computer science professor Michael Horn, computational literacy is the ability to use computers and computational technologies to solve problems, both in school and in the community. Computational literacy refers to having the knowledge that allows us to make use of the power of the computer beyond using typical applications.

REFERENCES

Century, J., Lach, M., King, H., Rand, S., Heppner, C., Franke, B., Westrick, J. (2012). Building an Operating System for Computer Science. Chicago, IL: CEMSE, University of Chicago with UEI, University of Chicago. Retrieved January 20, 2015 from http://outlier.uchicago.edu/computerscience/OS4CS/.

Computational literacy:

DiSessa, A. (2000). *Changing minds: Computers, learning, and literacy*. Cambridge: MIT Press.

Grover, S., and Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43.

Sherman, M. (2011). Computational literacy: As essential as ABC? *Inquiry.* The School of Education and Social Policy, Northwestern University. Retrieved January 20, 2015 from http://www.sesp.northwestern.edu/news-center/inquiry/2011-spring/computational-literacy.html

Computational thinking:

Aho, A. (2012). Computation and computational thinking. *The Computer Journal, 55*(7), 832-835. http://comjnl.oxfordjournals.org/content/55/7/832.abstract

Barr, V., and 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. http://csta.acm.org/Curriculum/sub/CurrFiles/BarrStephensonInroadsArticle.pdf

International Society for Technology in Education and the Computer Science Teachers Association (2011). *Computational thinking: Teacher resources.* Second Edition. Retrieved January 20, 2015 from http://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf (Definitions on pages 8-9).

Committee for the Workshops on Computational Thinking; National Research Council (2010). *Report of a workshop on the scope and nature of computational thinking.* The National Academies. Retrieved January 20, 2015 from http://sites.nationalacademies.org/cstb/CurrentProjects/CSTB_043590.

Committee for the Workshop on Computational thinking; National Research Council (2011). *Report of a workshop of pedagogical aspects of computational thinking*. The National Academies. Retrieved January 20, 2015 from http://sites.nationalacademies.org/cstb/CurrentProjects/CSTB_043590.

Computing at School Working Group (2012). C*omputer science as a school subject: Seizing the opportunity.* Retrieved January 22, 2015 from http://www.computingatschool.org.uk/data/uploads/Case%20for%20Computing.pdf.

Google resources for education (n.d.). *Exploring Computational Thinking.* Retrieved January 23, 2015 from http://www.google.com/edu/computational-thinking/what-is-ct.html.

Grover, S., and Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43. http://people.cs.vt.edu/~kafura/CS6604/Papers/CT-K12-Review-State-Of-Field.pdf

International Working Group (IWG) on Computational Thinking (2010). *Computational thinking for youth.* Education Development Center, Inc., Newton, MA.

Royal Society. (2012). *Shut down or restart: The way forward for computing in UK schools.* Retrieved January 22, 2015 from http://royalsociety.org/education/policy/computing-in-schools/report/.

Wing, J. (2014). *Computational thinking with Jeannette Wing*. Retrieved January 20, 2015 from http://aihub.net/computational-thinking-jeannette-wing/.

Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–36.

Computational thinking practices:

Grover, S., and Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43. http://people.cs.vt.edu/~kafura/CS6604/Papers/CT-K12-Review-State-Of-Field.pdf

John Carroll University (n.d.). *Computational thinking practices*. Retrieved January 22, 2015 from https://sites.google.com/a/jcu.edu/mt513/resources/cs-principles.

The College Board (2014). *Computer science principles*. Retrieved January 22, 2015 from http://media.collegeboard.com/digitalServices/pdf/ap/comp-sci-principles-draft-cf-final.pdf.

The College Board (2011). *Computer science principles*. Retrieved January 22, 2015 from http://www.collegeboard.com/prod_downloads/computerscience/ComputationalThinkingCS_Principles.pdf.

Computer engineering:

Joint Task Force for Computing Curricula 2005 (2005). *Computing curricula 2005: The overview report.* A joint project of the Association for Computing Machinery, the Association for Information Systems, and the Computer Society of the IEEE, September, 2005. http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf

University of Maine Computer Science (n.d.). *Computer science and computing-related fields: What are they and what's the difference?* Retrieved January 21, 2015 from http://umaine.edu/cs/what-is-cs/#ComputerEngineering.

Computer literacy:

Maryland Technology Literacy Standards For Students (n.d.). *Computer literacy skills: A companion to the Maryland literacy standards for students*. Retrieved January 22, 2015 from http://mdk12.org/instruction/curriculum/technology_literacy/computerliteracyskills.pdf.

National Center for Women & Information Technology. (2009). *Moving beyond computer literacy: Why schools should teach computer science.* Retrieved January 20, 2015 from http://www.ncwit.org/sites/default/files/resources/talkingpoints_curriculumandcs_webdownload.pdf.

PC Encyclopedia (n.d.). *Definition of: computer literacy*. Retrieved January 21, 2015 from http://www.pcmag.com/encyclopedia/term/40158/computer-literacy.

Wikipedia (n.d.). *Computer literacy*. Retrieved January 21, 2015 from http://en.wikipedia.org/wiki/Computer_literacy.

Computer science:

Ashcraft, C., Eger, E., Friend, M. (2012). *Girls in IT: The facts.* NCWIT. Retrieved January 22, 2015 from http://www.ncwit.org/sites/default/files/resources/girlsinit_thefacts_fullreport2012.pdf.

Computing at School Working Group (2012). C*omputer science as a school subject: Seizing the opportunity.* Retrieved January 21, 2015 from http://www.computingatschool.org.uk/data/uploads/Case%20for%20Computing.pdf.

Computing at School (2010). *A national working group.* Retrieved January 21, 2015 from http://www.computingatschool.org.uk/data/uploads/Computing_at_School.pdf.

Computing in the core (n.d.). *Understanding computer science education, information technology and technology literacy.* Retrieved January 21, 2015 from http://tinyurl.com/m72g7tn.

Computing in the core and Code.org (n.d.). *What is computer science and what do people do once they know it?* Retrieved January 21, 2015 from https://code.org/files/computer_science_is_foundational.pdf.

Computer Science Teachers Association and Association for Computing Machinery (2013). *Bugs in the system: Computer science teacher certification in the U.S.* New York: ACM. http://csta.acm.org/ComputerScienceTeacherCertification/sub/CSTA_BugsInTheSystem.pdf.

Computer Science Teachers Association Standards Taskforce (2011). *CSTA K-12 computer science standards.* New York: ACM. http://csta.acm.org/Curriculum/sub/K12Standards.html.

CSReq.org *(n.d.). What is computer science*? Retrieved January 22, 2015 from http://people.ucls.uchicago.edu/~bfranke/csreq/whatiscs.html.

DiSalvo, B., Reid, C., and Roshan PK. (2014). They can't find us: The search for informal CS education. Proceedings from the: *Special Interest Group Computer Science Education Conference, 2014*. New York, NY: ACM. http://dl.acm.org/citation.cfm?doid=2538862.2538933.

Gal-Ezer, J., and Harel, D. (1998). What (else) should CS educators know? *Communications of the ACM, 41*(9), 77-84.

Gal-Ezer, J., and Stephenson, C. (2010). Computer science teacher preparation is critical. *ACM InRoads, 1*(1), 61-66.

International Society for Technology in Education and the Computer Science Teachers Association (2011). *Computational thinking: Teacher resources.* Second Edition. Retrieved January 20, 2015 from http://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf (Definitions on pages 8-9).

Kaczmarczyk, L., and Dopplick, R. (2014). *Rebooting the pathway to success: Preparing students for computing workforce needs in the United States.* Association for Computing Machinery. Retrieved January 22, 2015 from http://pathways.acm.org.

National Center for Women & Information Technology. (2009). *Moving beyond computer literacy: Why schools should teach computer science.* Retrieved January 20, 2015 from http://www.ncwit.org/sites/default/files/resources/talkingpoints_curriculumandcs_webdownload.pdf.

Tucker, A. (2006). *A model curriculum for K-12 computer science: Final report of the ACM K-12 task force curriculum committee.* Second Edition. Retrieved January 22, 2015 from http://www.csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2ndEd.pdf.

Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–36.

Computing:

Ashcraft, C., Eger, E., Friend, M. (2012). *Girls in IT: The facts.* NCWIT. Retrieved January 22, 2015 from http://www.ncwit.org/sites/default/files/resources/girlsinit_thefacts_fullreport2012.pdf.

Association for Computing Machinery (n.d.). www.acm.org.

Computing at School (2010). *A national working group.* Retrieved January 21, 2015 from http://www.computingatschool.org.uk/data/uploads/Computing_at_School.pdf.

Joint Task Force for Computing Curricula 2005 (2005). *Computing curricula 2005: The overview report.* A cooperative project of the Association for Computing Machinery, the Association for Information Systems, and the Computer Society of the IEEE. Retrieved January 22, 2015 from http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf.

Cyberlearning:

Montfort, D., & Shane, B. (2013). What do we mean by cyberlearning: Characterizing a socially constructed definition with experts and practitioners. Journal of Science Education & Technology. 22(1): 90-102.

NSF Task Force on Cyberlearning (2008). Networked world: The cyberlearning opportunity and challenge. National Science Foundation. http://www.nsf.gov/pubs/2008/nsf08204/nsf08204.pdf

Digital literacy:

Jones-Kavalier B., & Flannigan, S. (2006). Connecting the digital dots: Literacy of the 21st century. *Educause Quarterly, 2*, 8-10. http://net.educause.edu/ir/library/pdf/eqm0621.pdf

University of Illinois University Library (n.d.). *Digital literacy definition and resources: What is digital literacy?* Retrieved January 22, 2015 from http://www.library.illinois.edu/diglit/definition.html.

US Digital Literacy (n.d.). http://digitalliteracy.us/.

Educational technology:

Computing in the core (n.d.). *Understanding computer science education, information technology and technology literacy.* Retrieved January 21, 2015 from http://tinyurl.com/m72g7tn.

Computer Science Teachers Association and Association for Computing Machinery (2013). *Bugs in the system: Computer science teacher certification in the U.S.* New York: ACM. http://csta.acm.org/ComputerScienceTeacherCertification/sub/CSTA_BugsInTheSystem.pdf

Computer Science Teachers Association Standards Taskforce (2011). *CSTA K-12 computer science standards.* New York: ACM. http://csta.acm.org/Curriculum/sub/K12Standards.html

Informatics:

Indiana University-Purdue University Indianapolis School of Informatics and Computing (n.d.). *Informatics defined*. Retrieved January 23, 2015 from http://soic.iupui.edu/about/what-is-informatics/.

University of Edinburgh Informatics (n.d.). *What is informatics?* Retrieved January 23, 2015 from http://www.ed.ac.uk/polopoly_fs/1.40075!/fileManager/What 20is 20Informatics.pdf.

Information and Communication Technology:

Computing at School Working Group (2012). C*omputer science as a school subject: Seizing the opportunity.* Retrieved January 21, 2015 from http://www.computingatschool.org.uk/data/uploads/Case 20for 20Computing.pdf.

Computing at School (2010). *A national working group.* Retrieved January 21, 2015 from http://www.computingatschool.org.uk/data/uploads/Computing_at_School.pdf.

TechTerms.com (2010, January 4). *ICT*. Retrieved January 23, 2015 from http://www.techterms.com/definition/ict.

UNESCO (n.d.). *ICT in education*. Retrieved January 23, 2014 from http://www.uis.unesco.org/Communication/Pages/ict-education.aspx.

Information science:

Cornell University InfoSci (n.d.). Retrieved January 23, 2015 from http://infosci.cornell.edu/.

University of Otago School of Business (n.d.). *About us: What is information science?* Retrieved January 23, 2015 from http://infosci.otago.ac.nz/what-is-information-science/.

Information Systems:

Joint Task Force for Computing Curricula 2005 (2005). *Computing curricula 2005: The overview report.* A cooperative project of the Association for Computing Machinery, the Association for Information Systems, and the Computer Society of the IEEE. Retrieved January 22, 2015 from http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf.

Information Technology:

Ashcraft, C., Eger, E., Friend, M. (2012). *Girls in IT: The facts.* NCWIT. Retrieved January 22, 2015 from http://www.ncwit.org/sites/default/files/resources/girlsinit_thefacts_fullreport2012.pdf.

Computing in the Core (n.d.). *Understanding computer science education, information technology and technology literacy.* Retrieved January 21, 2015 from http://tinyurl.com/m72g7tn.

Computer Science Teachers Association and Association for Computing Machinery (2013). *Bugs in the system: Computer science teacher certification in the U.S.* New York: ACM. http://csta.acm.org/ComputerScienceTeacherCertification/sub/CSTA_BugsInTheSystem.pdf.

Computer Science Teachers Association Standards Taskforce (2011). *CSTA K-12 computer science standards.* New York: ACM. http://csta.acm.org/Curriculum/sub/K12Standards.html

TechTerms.com (n.d.). *IT*. Retrieved January 23, 2015 from http://www.techterms.com/definition/it.

Programming and coding:

Beal, V. (n.d.). *Code*. Retrieved January 26, 2015 from http://www.webopedia.com/TERM/C/code.html.

Bitautas, J. (2013, March 11). The differences between programmers and coders. Retrieved January 26, 2015 from http://workfunc.com/differences-between-programmers-and-coders/.

BusinessDictionary.com (n.d.). *Computer programming*. Retrieved January 26, 2015 from http://www.businessdictionary.com/definition/computer-programming.html.

Suggested citation: LeadCS.org (2015). Computer Science Terminology. CEMSE, Outlier Research & Evaluation, University of Chicago. http://www.leadCS.org