# Importance of Mathematics in Computer Science, Engineering and Technology (CSET) Education

Francis Avevor
*Advanced School of Data and System Studies, Accra Institute of Technology, Accra, Ghana*

***Abstract*: Students with weak mathematics background are often overwhelmed by the many mathematics courses that they have to take during their undergraduate Computer Science, Engineering and Technology (CSET) programmes; and tend to question the relevance of mathematics to their CSET studies. This paper examined the syllabuses of undergraduate CSET courses being offered in 10 Ghanaian universities and discussed the importance of relevant mathematics courses to them. Furthermore, it proffered recommendations to promote students' academic success in CSET programmes.**

***Keywords*: Mathematics, CSET education, logical reasoning, algorithm analysis, problem solving.**

## I. INTRODUCTION

Mathematics is a discipline that "develops the ability to reason precisely and analytically about formally defined abstract structures" [1] p.39; and it complements the learning, understanding and appreciation of many theoretical frameworks as well as foundational concepts and algorithms in CSET education [2]. Specifically, mathematics teaches critical thinking and reasoning, as well as analytical skills for data analysis and problem-solving; and its concepts are required in many CSET disciplines [3]. Thus, many universities have incorporated several mathematics courses in their curriculum for CSET education. Generally, students who have strong mathematics background do well in the CSET courses and professions; while those with weak mathematics background struggle to cope with the CSET courses and professions. This is because Mathematics is intrinsically tied to the study of CSET courses at various levels of education [4], [5].

## II. PROBLEM STATEMENT

Undergraduate students in Computer Science, Engineering and Technology (CSET) are often overwhelmed with the many mathematics courses they are required to do in the first four semesters of their studies. Some of them tend to question the relevance of the many mathematics courses they take. Students who have weak mathematics background typically have difficulty in understanding concepts in CSET courses such as Data Structures and Algorithms, Computer Programming, Software Engineering, System Analysis and Design, Artificial Intelligence, and Network Security; and they tend to perform abysmally in these (and related) courses. This paper examined the importance of mathematics courses in undergraduate CSET education through a review of literature. The sole research question for this paper is: What is the importance of mathematics courses in undergraduate CSET Education?

The paper reviewed the syllabuses and recommended textbooks of undergraduate CSET courses in 10 Ghanaian universities to examine the importance of major mathematical courses that CSET students take. Table 1 presented the list of universities covered for this paper and their CSET programme offerings. It showed that University of Ghana (UG), Accra Institute of Technology (AIT), and Kwame Nkrumah University of Science and Technology (KNUST) are the three most dominant tertiary institutions offering a wide range of CSET programmes up to the PhD level in Ghana.

Table 1. Some Ghanaian Universities Offering Cset Programmes.

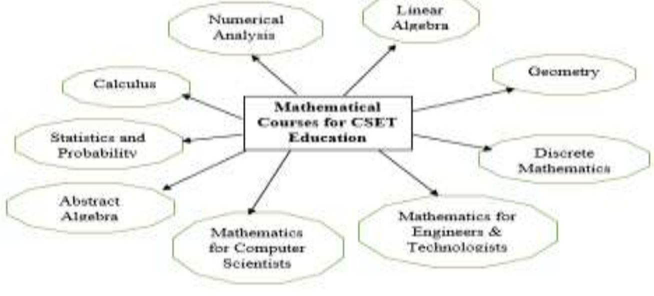
| Institution | CSET Programme Offerings |
|---|---|
| 1. University of Ghana, Legon | BSc/MSc/MPhil/PhD Computer Science, BSc/MSc/MEng/MPhil/PhD Computer Engineering, BSc/MPhil/PhD Information Technology, MBA/MPhil Management Information Systems, MPhil/PhD Nuclear Engineering |
| 2. Kwame Nkrumah University of Science & Technology, Kumasi | BSc/MSc/MPhil/PhD Computer Science, BSc/MSc/MPhil/PhD Computer Engineering, BSc/MSc/MPhil Information Technology, MEng Information and Communications Technology, BSc/MSc/MPhil/PhD Electrical/Electronic Engineering, MPhil/PhD Telecommunications Technology, BSc/MSc/MPhil/PhD Mechanical Engineering |
| 3. Accra Institute of Technology | CTech/DTech/BSc/MSc/PhD Information Technology CTech/DTech/BSc/MSc Computer Science, MSc Software Engineering, MSc Multimedia Communications, BSc Civil Engineering, MSc/PhD Engineering, MSc Environmental Science |
| 4. Regent University College, Accra | BSc Information Technology, BSc Computer Science, BEng Computer Engineering, BEng Telecommunications Engineering, BEng Instrumentation Engineering. |
| 5. Ashesi University, Accra | BSc Computer Engineering, BSc Electrical/Electronic Engineering, BSc. Computer Science, BSc. Mechanical Engineering, MSc/MPhil/MAS Mechatronic Engineering, BSc Management Information Systems |
| 6. Central University | BSc Computer Science, BSc Information Technology, BSc Civil Engineering. |
| 7. Accra Technical University | HND/BTech Computer Science, BTech Management Information Systems, HND/BTech Electrical/Electronic |

| | | |
|---|---|---|
| | | Engineering, HND/BTech Civil Engineering, HND/BTech Mechanical Engineering, BTech Telecommunications Engineering |
| 8. | University of Cape Coast | BEd/BSc/MEd Computer Science, BEd/BSc/MEd Information Technology |
| 9. | Ho Technical University | HND/BTech Computer Science, HND/BTech Information & Communication Technology, MTech Production Engineering, BTech Energy Engineering, HND Electrical & Electronic Engineering, BTech/MTech Automobile Engineering, HND/BTech/MTech Agricultural Engineering, HND Civil Engineering, HND Mechanical Engineering. |
| 10. | Catholic University College, Fiapre | Dip/BSc/BEd Computer Science, BSc/BEd Information Technology, BSc/BEd Computing & Business Analytics. |

## III. MAJOR MATHEMATICS COURSES FOR CSET EDUCATION

Figure 1 showed the major mathematical courses that undergraduate students take in their CSET education programmes in Ghanaian Universities. Specifically, the students usually take Discrete mathematics, Linear algebra, Geometry, Mathematics for Computer Scientists, and Mathematics for Engineers and Technologists during the first four semesters of their undergraduate CSET programmes. These mathematical courses provide the foundation for understanding the concepts of many CSET courses that students take in their first two years of undergraduate studies such as Computer programming, Data structures and algorithms, Computer organization and architecture, and Compiler design and construction. Furthermore, students also take more advanced mathematical courses such as Numerical analysis, Calculus, Abstract algebra, and Statistics & probability during the $5^{th}$ - $8^{th}$ semesters of their programmes. These courses provide students with the basis for more rigorous CSET courses like Artificial intelligence and machine learning, Computer networks and data communications, Theory of computation, and Network security. Every undergraduate CSET course requires logical reasoning, critical analysis and real-life problem-solving. For students to be able to cope with the CSET courses, they use their critical thinking skills as well as their knowledge of concepts, logic and proofs from mathematical courses such as discrete mathematics, linear algebra, abstract algebra, and geometry.

Figure 1. Major Mathematical Courses for CSET Education




## IV. OVERVIEW OF MAJOR CSET EDUCATION COURSES THAT REQUIRE STRONG BACKGROUND IN MATHEMATICS

This section of the paper provides an analysis of major CSET courses that require students to have a strong mathematics background. The analysis will reveal the importance of mathematics courses in CSET education, in line with the sole research question (RQ1).

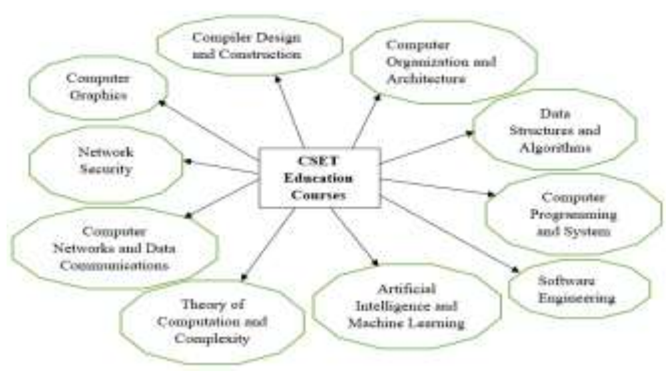
*RO1.What is the importance of mathematics courses in undergraduate CSET education?*

Figure 2 showed ten major CSET courses that Universities in Ghana require their CSET students to have strong mathematics backgrounds in, in order to succeed in their undergraduate studies and future careers in CSET fields. These CSET courses are analyzed as follows:

### *A. Compiler Design and Construction*

Students learn to design, construct and implement compilers for various programming languages to foster efficient translation of source code, written in high-level programming languages, into executable object code. Students learn concepts such as formal languages, program analysis (comprising lexical analysis, syntax analysis, and semantic analysis), automata, parsing, intermediate and target code generations, debugging, and code optimization. For students to understand, appreciate and apply these concepts, they typically take mathematical courses such as discrete mathematics as well as linear and abstract algebra. Compiler design and construction requires critical analysis and logical reasoning; and for students to be able to cope with this course, they have to use their knowledge of concepts, logic and proofs from mathematical courses such as discrete mathematics, linear algebra, abstract algebra, and geometry.

Figure 2. Major CSET Education Courses that Require Mathematics




### *B. Computer Organization and Architecture*

This course equips students with knowledge and skills for designing, organizing, constructing, verifying and optimizing computer systems cost-effectively as well as analyzing system performance issues. Typical topics for this course include instruction set, addressing, memory optimizations, data conversions, pipelining, parallelisms, synchronization and system performance metrics (relating to throughput and latency). These topics, therefore, require students' knowledge in discrete mathematics (notably: Boolean algebra, circuit design, and number systems). Also, students use their knowledge from calculus (both differential calculus and integral calculus) to understand the concepts and operations of computer hardware electrons, circuits and systems. To be able to convert number systems from binary to octal, decimal and hexadecimal (and vice versa), and to successfully design device drivers and hardware circuits, students need knowledge from discrete mathematics (specifically, Boolean algebra and binary arithmetic).

*C. Data Structures and Algorithms*

In this course, students learn fundamental data structures (including stacks, queues, sets, lists, arrays, hash tables, trees, maps, and graphs), algorithm analysis, sorting and searching, and principles of language translation. These help them to develop their algorithmic skills and competencies. For students to understand the lessons in this course, they have to take discrete mathematics (specifically graph theory and proof techniques). Graph theory in discrete mathematics provides a vital foundation for searching and sorting algorithms, algorithm analysis (to determine the space and time complexity of algorithms) and for data structures. For students to understand, appreciate and implement binary-search-tree-based sets and hash-tables-based sets in this course, they need knowledge from set theory (such as union, intersection, and difference of sets) and graph theory. [6] assets that undergraduate data structures and algorithms course requires "strong Mathematical skills such as abstraction ability and algorithmic thinking but do not require strong rigorous mathematical reasoning skill" (p. 3).

*D. System Analysis and Computer Programming*

In this CSET course, students acquire knowledge and understanding of computer programming principles, practices and tools for C++, Visual Basic, Java and Phyton. They also acquire computational problem-solving skills, communication skills, cognitive skills, and project management skills. They learn topics such as algorithm analysis, simulation, software quality metrics, code optimization, dynamic storage allocation principles, parallelism and parallelization techniques, and computational complexity of algorithms. Students' knowledge in calculus, discrete mathematics, abstract algebra, geometry, and linear algebra are necessary for them to understand this course. Specifically, discrete mathematics, geometry and linear algebra help them to understand graph algorithms, coding theory, logical programming, proofs of program correctness, error correcting codes generation, and pseudo randomness. Also, calculus is relevant for performing simulations, and for algorithms design and analysis.

Most of the ten universities reviewed for this paper teach students programming in C++, CSS, Phyton, Java and VB.Net. Variables declaration and use in these programming languages require students' knowledge about variables from their linear algebra and discrete mathematics lessons. Also, students need their knowledge on functions and data types to perform efficient analysis of code performance regarding functions, and for determining the adequacy of test data that they use. They could utilize their knowledge of mathematical functions by expressing operations in their programs as mathematical functions. Furthermore, mathematical expressions that students learn in discrete mathematics and linear algebra are relevant for efficient coding, program testing and debugging of programs that they write. Specifically, for students to successfully debug their programs, they need knowledge from mathematical proofs and formal languages so as to precisely determine appropriate language definitions and their abstracts in terms of both syntax and semantics.

Furthermore, computer programming course requires critical problem/system analysis and logical reasoning (during the analysis, design, coding, testing and implementation phases). For students to be able to cope with this course, they need to use their critical thinking skills as well as their knowledge of mathematical concepts, logic and proofs from discrete mathematics, linear algebra, abstract algebra, and geometry. These days, many mobile application developers are developing various navigational and tracking systems to support supply-chain activities. In one of my Advanced System Analysis and Design classes a few semesters ago, I requested my students to analyze and design a system in this regard; and they skillfully utilized their knowledge of discrete mathematics, abstract algebra, calculus, statistics and geometry in order to perform the system analysis, specify the algorithms, and design the system.

*E. Software Engineering*

This CSET course is "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software" [7], p.25. It focuses on software estimation, analysis, design, testing and quality improvement [8]; and it "integrates significant mathematics, computer science and practices whose origins are in engineering" [9], p. v). In this undergraduate CSET course, therefore, students acquire knowledge, skills and tools for tasks spanning all the five phases of the software development life-cycle: system analysis, design, coding, testing, and maintenance. Some of the topics that students learn in this course include principles and practice of software quality metrics, software testing, proofs of program correctness, software project management, as well as code complexity and optimization. Their knowledge in discrete mathematics, calculus, geometry, and linear algebra are very necessary for them to fare in this course. For instance, they need to use mathematical approaches to precisely and accurately describe, analyze and design software. To develop good quality software in this course, students need to utilize their knowledge in logical reasoning, Boolean algebra, set theory, mathematical expressions and functions, and mathematical proofs, among others. The diagrams that CSET students and professionals draw for planning and controlling of software projects (such as project management workbench, and program evaluation and review techniques (PERT)) require a good mathematical background involving graph theory.

*F. Artificial Intelligence (Ai) And Machine Learning (Ml)*

Students learn data mining, speech recognition, data compression, vision and image analysis, and simulation, among other relevant topics in this CSET course. Number theory and mathematical logic provide them with the logical reasoning and abstraction they need to foster their understanding in this course. Mathematical courses that are specifically important for this course are discrete mathematics, calculus, linear algebra, geometry, abstract algebra, statistics and probability. Specifically, for students to design and construct relational databases, they require knowledge of graph theory from discrete mathematics. Similarly, for students to create axioms and rules for machine learning operations, they require mathematical logic and proofs from discrete mathematics and abstract algebra, respectively. Students also use their knowledge from statistics and probability to conduct empirical research in AI and ML. Also, for students to efficiently generate visual images and then integrate visual and spatial information, they need to use their knowledge in graph theory from discrete mathematics and geometry. Furthermore, for students to understand data representation and abstraction, generation of various design patterns (such as structural design patterns, computational design patterns, implementation patterns, and execution patterns), encapsulation, systematization and polymorphism in data mining, they require necessary knowledge from discrete mathematics, linear algebra, and geometry [10], [11].

*G. Theory of Computation and Complexity*

In this CSET course, students typically learn topics such as syntax analysis, automaton, complexity, game theory, logical theories, optimization theory, time and space complexity, discrete theory and structures, NP-completeness, NL-completeness, primality testing, recursion theorem, control theory and cryptography. These topics rely heavily on mathematical courses such as geometry, calculus, probability, linear algebra and abstract algebra. Thus, undergraduate CSET students need knowledge from these mathematical courses to understand concepts, tools and methods in the theory of computation and complexity. This undergraduate Theory of Computation and Complexity course, therefore, requires students to have "strong rigorous mathematical reasoning ability together with other mathematical abilities to learn, understand, and appreciate the theories". [6].

*H. Computer Networks and Data Communication*

Computer networks are often designed and implemented to solve various problems such as routing issues, security issues, network load issues, recovery issues and efficiency-related issues. Undergraduate students require knowledge in graph theory and transformations (from discrete mathematics and geometry), to understand concepts and practices relating to these network issues and then design and implement efficient, effective and innovative world-class solutions for them. Also, students need knowledge from statistics and probability to perform empirical analysis of computer networks and data communication issues. Furthermore, for students to understand and appreciate networking protocols and algorithms for data communication and security, they need knowledge from linear algebra, discrete mathematics and abstract algebra.

*I. Computer Security*

Students also need knowledge from abstract algebra (especially mathematical logic and proofs) to understand various data compression and encryption algorithms that are used in computer networks and systems**.** They also need knowledge from statistics and probability to perform empirical analysis of computer security issues. Cryptography, in general, requires knowledge of mathematical courses such as discrete mathematics, probability, linear and abstract algebra, and calculus. Specifically, for students to understand algorithms for cryptographic key generation (such as DES, AES, and RSA), digital signing and verification, they need to have knowledge in mathematical logic and proofs from abstract algebra and discrete mathematics.

*J. Computer Graphics*

In undergraduate computer graphics course, students learn the algorithms underlying graphics-intensive applications, 3D and 2D graphics, homogenous coordinates, scan conversions, illumination, 3D viewing and transformations, clipping, and OpenGL programming. These topics rely heavily on foundations of algorithms and basic mathematical concepts (such as set theory, proof techniques and graph theory) in linear algebra, discrete mathematics and abstract algebra. Thus, students need to have a good background in these mathematical courses to understand, appreciate and cope with the Computer Graphics course.

## V. CONCLUSION AND RECOMMENDATIONS

This paper has analyzed the importance of mathematics courses for CSET education and it has shown that many undergraduate CSET courses require students' knowledge in relevant topics from a number of mathematics courses. Invariably, students with a weak mathematics background find it difficult to cope with CSET courses at the undergraduate level. They often ask spurious questions in class showing their lack of, or inadequate grasp of, basic mathematics concepts necessary for CSET education. This tends to put enormous pressure on lecturers to find innovative strategies to make such students cope with their CSET courses. It is, therefore, important that students complete a sufficient number of mathematics courses prior to their registration for various undergraduate CSET courses.

This paper proffers the following recommendations to increase students' understanding and appreciation of CSET courses:

i. Universities providing CSET education should offer all relevant mathematics courses.
ii. Lecturers should make their CSET classes student-oriented, lively and practical problem-solving environments.
iii. Lecturers should design and tailor their CSET classes to the needs of students with weak mathematics background. Teaching assistants should provide extra tutorial sessions on mathematical topics for them.
iv. Since "practice makes perfect" [12], p. 105, weekly assignments should be given to the students to do in each CSET course to enable them to apply what they have learnt in class.

Furthermore, this paper proffers the following recommendations for future research considerations:

i. Future research may consider performing a comparative analysis of the impact of mathematics courses on the performances of undergraduate Computer Science students, Engineering students and Information Technology students in CSET courses.
ii. Researchers may consider focusing on determining the critical success factors for enhancing undergraduate students' performance in CSET courses.
iii. With the current raging debate on the adequacy or otherwise of the number of mathematics courses that a university should request its students in CSET programmes to take, future research may consider conducting an in-depth empirical literature review on the current debate for increasing or reducing the number of mathematics courses in the curricula of CSET programmes and proffer cogent recommendations to university administrators.

ACKNOWLEDGMENT

I acknowledge Evans Kudzo Morve for editing the manuscript for this study.

REFERENCES

[1] Devlin, K. (2003). Why Universities Require Computer Science Students to take Math. Communications of the ACM, 46(9), 37-39
[2] Henderson, P. B. (2005). The role of mathematics in computer science and software engineering education. Advances in Computers, 65: 350-396.
[3] Su, H.F.H., Ricci, F.A., and Mnatsakanian, M. (2016). Mathematics Teaching Strategies: Pathways to Critical thinking and Metacognition. International Journal of Research in Education and Science, 2(1), 190-200.
[4] Glass, R.L. (2000). A New Answer to How Important is Mathematics to the Software Practitioner? IEEE Software. 135-136
[5] Beaubouef, T. (2002). Why Computer Science Students Need Math. SIGCSE Bulletin, 34(4) 57-59.
[6] Sritharan, T. (2018). Impact of mathematics on the theoretical computer science course units in the general degree program in computer science at Sri Lankan state universities. Issues in Informing Science and Information Technology, 15, 1-14. https://doi.org/10.28945/4007
[7] Abran, A; Moore, J.W; Bourque, P; and Tripp, L.L. (2005). Guide to the Software Engineering Body of Knowledge. Los Alamitos: IEEE Computer Society Press.
[8] Singh, Y., and Malhotra, R. (2012). Object-Oriented Software Engineering. Delhi: PHI Learning Pvt Ltd.
[9] Wu, Y. (Ed.). (2012). Software Engineering and Knowledge Engineering: Theory and Practice: Volume 2. Verlag Berlin: Springer Science & Business Media.
[10] Fuggetta A. "A Classification of CASE Technology". Dipartimento di Elettronica ed Inf., Politecnico di Milano: Computer, vol. 26, no. 12, Dec. 1993, doi: 10.1109/2.247645. – pp. 25-38.
[10] Armstrong, D.J. (2006). The quarks of object-oriented development. Communications of the ACM. 49(2) 123-128. doi: 10.1145/1113034.1113040
[11] Fuggetta, A. (1993). A classification of CASE Technology. Computer. 26(12) pp 25-38. Doi: 10.1109/2.247645.
[12] Chae, S. (2001). The student guide to Computer Science C++. Bloomington: Writers Club Press.