IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 6.252
Volume 2, Issue 2, February 2022

# Mathematics: The Root of Computer Engineering

**Kiran B. Mate, Kaveri Suryawanshi, Vishakha Shirsath, Rani Nirmala**
Sanjivani K. B. P. Polytechnic, Kopargaon, Ahmednagar, Maharashtra, India

**Abstract:** *Math like every other field of life, matters for computer science and engineering because it teaches students how to use abstract language, work with algorithms, self-analyze their computational thinking, and accurately model real- world solutions. In the current world, the call to expand access to computer science education in schools is increasing. As the educational community focuses on this challenge, the relationship between math education and computer science is undoubtedly moving into the spotlight. This relationship can be a controversial subject. Part of this debate springs from differing ideas of what constitutes a "strong background in mathematics." For some, it's getting a Ph.D. mathematics. For others, it's being able to solve multistep equations mentally.*

**Keywords:** Abstract Language, Algorithm, Computational Thinking, Multistep Equations

## I. Introduction

From the early days of school final stages of graduation in engineering, Mathematics has always been a tough subject to study and the students are always struggling with a fear factor associated with mathematics. Although practice makes a man perfect hold truth in many typical tasks it also holds true for study as well. Mathematics has been too complicated for the majority of students. But mathematics as the primary subject in core and new engineering courses is compulsory in worldwide engineering education. In engineering, there are several branches such as computer science engineering, electrical engineering, mechanical engineering, electronics, communication engineering, and civil engineering and each branch has a different study set focused on the significance of mathematics. For Computer Engineers, Math matters because it teaches students how to use abstract language, work with algorithms, self-analyze their computational thinking, moderately real-world solutions.

## II. Reasons to Support Our Claim

There's another way of identifying a strong math background: having the capacity for abstract reasoning, critical thought, and logical deduction - a mathematical way of thinking. In this regard, a strong background in mathematics is imperative to succeeding in computer science and engineering.

### 1. Math teaches understanding and communication through abstract language:

Computer programming has its languages very abstract languages tax, one must represent specific processes, commands, and visuals through punctuation, symbols, and single words. To someone with no experience thinking or communicating in abstract languages, learning a programming language can be terrifying.

However, abstract programming languages are very similar to the mathematical language that students learn in math class. From simple equalities to complex mathematical representations, learning mathematics teaches students the art of reading, comprehending, formulating thoughts, and communicating with abstract language. Of course, mathematical language and computer programming languages aren't exactly the same. But experience using any abstract language gives beginning computer scientists an advantage.

### 2. Math teaches how to work with algorithms

The algorithm is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. The algorithm is among the most common term in the technology scene. In short, an algorithm is an abstraction of some process into a form in which the process can be repeated, implemented in different ways, and applied to new problems.

The word may be used more frequently in computer engineering, but most students first use algorithms in mathematics. For example, consider an equation like $3 + x = 5$.

Students learn to find an unknown summand by subtracting the known summand from the sum. This is an algorithm one that students quickly learn to apply to new problems and implement in different ways.

**3. Math teaches students how to analyze their work.**

In a day's world of programming, any computer engineer is guaranteed to make a mistake. As such, programmers must know how to assess their work, and fix errors. Mathematics is one of the few subjects where we analyze our work in this way. A student might answer a math question (such as How much do the apple and mango weigh together?), realize that their answer is unreasonable (15 KGs), and analyze their process to understand their mistake and how to fix it (maybe they forgot to convert from gms to kgs or pounds). Math, in short, prepares students for fixing bugs.

**4. General skills aside, computer science still involves a lot of math.**

In addition to general skills important for computer science, the facts and figures of math are essential. As computer programming interacts more with our world, the importance of accurately modeling that world through mathematics grows. For example, to build a self-driving car, the equations used to program its turns, acceleration, and acceptable distance from other cars must be spot-on.

Becoming a computer scientist requires a fair amount of mathematical knowledge and skill. Even more importantly, success in computer science requires the ability to think mathematically. So why is it necessary to talk about how math helps prepare budding computer scientists for their academic careers?

**5. Discrete Mathematics**

In discrete mathematics, you study objects that have distinct, separated values, such as integers, graphs, and statements in logic. It stays away from continuous mathematics topics like calculus. Instead, it includes those that can be used to answer various tangible inquiries. As such, discrete math is an important building block throughout computer science as its problem-solving methods help create complicated software. You'll see it used in computer architecture, algorithms, computer systems, databases, distributed systems, machine learning, operating systems, and more.

**6. Algebra**

In algebra, you learn to calculate formulas and equations to find variables or solutions for both simple and complex problems. By learning this subject, you learn to define relationships between objects and problem solve with limited variables. Which enhances your analytic ability. In computer science, algebra is used in many ways. From evaluating code paths to processor optimizat ion, and relational database design. But it also plays a big part in the development of algorithms and software working with mathematical objects. As well as designing formulas used within numerical programs and for complete scientific computations.

**7. Statistics**

By studying statistics, you look at ways to obtain, review, evaluate and draw conclusions from data. Utilizing a diverse range of measurement methods like mean, variance, analysis of variance, and kurtosis to paint a clearer picture. In computer science, statistics plays a fundamental part. As it is used throughout areas like data mining, speech recognition, vision and image analysis, and even artificial intelligence. In learning this subject, you'll better understand algorithms and other important aspects of computer science.

**8. Calculus**

Through calculus, you learn about continuous change and the rates change occurs. Both types of calculus (integral and differential) will enable you to understand the rate of change of a quantity. In computer science, you'll program systems to monitor and even predict change. Whether this is in algorithms, simulations, or problem-solving applications, understanding differential equations will help your work.

**9. Binary Mathematics**

Computers use the digits 0 and 1 to store data, otherwise known as binary. That puts binary math at the heart of computer operations, making it likely the most important math subject to learn. It is used to symbolize every number within a computer and acts as an alternative to the decimal system. The reason for this is it helps simplify computer designs, reading, and mathematical operations for hardware low-level programming.

A hexadecimal (or hex) number system is a base 16 system that is used to simplify how binary is represented. Providing a user-friendly representation of binary-coded values. So, knowing how to work with this is necessary to perform programming functions such as setting colors. You'll use standard arithmetic in many functions of computer programming. For instance, addition, subtraction, multiplication, and division are used in almost every written program.

## III. Other Importance

Mathematical models and algebra are involved and equally interconnected in handling all the database queries of complex algorithmic and working on websites. If the programmers are working on high- end technologies such as machine learning and artificial intelligence computer programmers should have sola id foundation in maths subjects such as Statistics, Calculus, Probability in engineering mathematics theory. Core mathematical topics such as discrete mathematics syllabus including Set theory, Graph theory, Probability, Number theory, Algebra is the basic foundation for programming and computer engineering. Cryptography programming is logically based and mathematics Number theory has multiple applications in cryptography, on the other hand, we have Boolean algebra and relational algebra which are used in logic gates and databases.

## IV. Conclusion

Becoming a computer engineer requires a fair amount of mathematical knowledge and skill. Even more importantly, success in computer studies requires the ability to think mathematically. A strong math background develops all of the skills mentioned above. Unfortunately, as it's currently taught in our education system, math education doesn't always develop a strong math background in its students. Many math teachings and classes focus on the rote memorization of formulas. These classes neglect to build the critical thinking and logical reasoning that help students in future math classes and computer science careers.

## Acknowledgment

Math is an essential component of computer scientistic underpins computing and programming concepts. Without it, you would find it challenging to make sense of abstract language, algorithms, data structures, or differential equations. All of which are necessary to fully appreciate how computers work.

Authors are thankful to everyone involved viz the co-authors, the teachers, the faculties of Guru Gobind Singh Polytechnic and especially Prof. G.R.Jagtap, Prof. O.A.Husain, and Prof. V.N.Pawar. The authors are highly grateful to Respected Principal Guru Gobind Singh Polytechnic, Hon. Executive Director of Guru Gobind Singh Foundation, Nashik for the opportunity and timely guidance and help provided by them.

## References

**[1].** Mathematics for Computer Science by Eric Lehman, F. Thosmon & Albert R. Meyer
**[2].** Mathematics in Computer Science, Christopher J. Van Wyk, Instt. Of Computer Science and Technology, NBS, Washington D.C. 20234
**[3].** The role of mathematics in Computer Science by Doug Baldwin
**[4].** Mathematical applications in Computer Science: A Review by A.K.Saxena, Dr. Saurabh Saxena
**[5].** The Reader's blog through stonebridge
**[6].** Knuth, D. E., The Art of Computer Programming Vol. 2
**[7].** Seminumerical algorithms, Addison Wesley, New York, 1981.
**[8].** Yan, S. Y., Number Theory for Computing, Springer, New York, 2000.
**[9].** Knuth, D. E., The Art of Computer Programming Vol. 3, Sorting and Searching, Addison Wesley, New York