Calculus Education · Computer Programmers · Calculus · Computer Programming

As a CS student, why do I need to take calculus when no programmer will ever need to use it?

Answer · Follow · 33 · Request

62 Answers

Asked in 1 Space

Matthew L Reigada, Software Engineer (2014-present)
Updated September 12, 2018 · Upvoted by Alan Mellor, Started programming 8 bit computers in 1981 and Paul Barnes, MS Applied Mathematics & Computational Science and Scientific Computing, University of Tulsa (1997)

As a CS student, why do I need to take calculus when no programmer will ever need to use it?

I'm gonna answer this in multiple parts:

1. Computer Science =/= Computer Programming
2. Why specifically an understanding of Calculus is useful in CS
3. Why this is the wrong attitude with which to approach Math, Calculus, Computer Science, and knowledge

Computer Science =/= Computer Programming:

- There's a very significant difference in talent and pay between a *mechanic* and a *mechanical engineer*
  - A *mechanic* might install engines in a car
  - A *mechanical engineer* might design engines and specify how they are to be installed by mechanics into cars
  - A *mechanical engineer* who designs engines, still needs to know how engines function AND how they are typically installed in order to create their product
- There's a very significant difference in talent and pay between a *electrician* and a *electrical engineer*
  - An *electrician* might install wiring through a house
  - An *electrical engineer* might design a power-grid and specify how houses are to be integrated by electricians into that power grid
  - An *electrical engineer* who designs power-grids, still needs to know how power-grids function AND how houses and businesses are typically integrated within power-grids in order to create their product
- Programming, although perhaps the most important skill, is still only one of many skills that Computer Science entails
  - A *programmer* might only need to know a JS framework or two to know how to work in the front end professionally
  - A *computer scientist* or *software engineer* might need to know:
    -How a client-server architecture functions
    -How internet architecture functions more generally
    -How programming languages function
    -How to manage a CPU-bound workload
    -How to manage an IO-bound workload
    -How hardware interfaces with software
    ...
- If *only* programming (as a vocation) is your end goal then you are likely either:
  - Setting your bar too low and you should aim higher
    -OR-

bootcamp for a tenth of the time and price of a degree

- I've personally worked with perfectly competent front-end developers and mobile developers who have educational backgrounds in anything from Mandarin Literature to Biology and Nursing. This is a perfectly viable career route with lowered technical expectations, but you will be paid less. If you go this route with a CS degree then you probably wasted your degree and could have studied something you more directly enjoyed.

Why specifically an understanding of Calculus is useful in CS:

- *"... when **NO** programmer will ever need to use it?"*
  ^Use of an absolute is entirely incorrect here^
  - *"**Many** programmers won't ever put calculus to use"* <- this is true
  - How useful Calculus is will be entirely up to you and your use of it, if you never use it then its never useful.
  - A hammer that a carpenter never swings is a useless hammer, but that doesn't mean that hammers are useless to carpenters in general and that you as a novice carpenter should actively plan to never swing one
- Machine Learning uses Calculus all the time
- Image Processing uses Calculus all the time
- Signal Processing uses Calculus all the time
- Computational Complexity uses Calculus all the time
- Data Mining uses Calculus all the time
- Aviation Software uses Calculus all the time
- Game Engine Programming uses Calculus all the time
- Computational Statistic uses Calculus all the time
- Scientific Computing uses Calculus all the time
- [ANOTHER_HUNDRED_APPLICATION_SPACES] uses Calculus all the time
- When one personally understands the root concepts of Calculus, it fundamentally changes their way of thinking
  - Even when you aren't explicitly using Calculus, the vague meaning of a phrase like "*diminishing returns*" isn't vague at all, it is explicitly quantifiable
  - In every problem that requires optimization, you have an explicit mathematical toolbox for handling the problem
  - When handed a document with an arcane mathematical formula related to your problem space, you can not only discern the literal formula, but how to implement it in code, and why that formula applies to the problem space

Why this is the wrong attitude with which to approach Math, Calculus, Computer Science, and knowledge:

Mathematics in general, but Calculus especially, is one of those things where the layman will say it's not useful because it is very difficult to learn. At a subconscious level we are all aware that if something is useless, then we don't have to be bothered to waste energy learning it. So by saying "*this is useless, I don't need to learn it*" you are making a career choice that you will keep this tool out of your toolbox and will remain non-proficient with this tool. There is no knowledge that will harm your personal or professional perspective, only knowledge that is less helpful than other knowledge.

If you have done so because you have finite time resources and there are more immediately useful tools to shove into that toolbox then all the more power to you. There may in fact be tools far more proximal to your application space. However, if you are omitting this tool from your toolbox simply because it's hard to learn then you are cutting yourself at the knees professionally.

You are in school right now. You are literally doing nothing but making an effort to learn. As long as that remains the case, it utterly behooves you to learn the most difficult but useful tools, as now is the best time to focus on learning them and since they are difficult you will have a competitive edge over peers who did not make the

minimum you will have proven that you can learn something absolutely difficult, and that shows that you can learn other things with ease.

Bonus to that point: I once approved a hire for a back-end developer position. After approving the hire I was asked why I endorsed the candidate when he had almost none of our requisite tech stack listed in his experience. My answer: we had a niche tech-stack on that project and we'd never find an ideal candidate with that experience. I approved this guy for hire because he went off on a tangent mid-interview just to showcase a non-trivial project he made in Brainfuck, not because it was a practical language but just because he wanted to learn something difficult for fun. He was straight out of school, but he was one of the top devs inside of 6 months. It's not just important to show that you know the right tool for the job, but also important that you can show willingness to learn new tools or invent your own with ease in those situations you need to (but don't look for excuses to reinvent the wheel).

118.5K views · View 1,145 upvotes · View shares

1.1K 37 38

38 comments from E Douglas Jensen and more

Fred Mitchell, An old Autodidact at Software Engineering & Comp Sci.
Answered November 25, 2020

As a CS student, why do I need to take calculus when no programmer will ever need to use it?

What makes you think you'll never need to use it?

One of my recent jobs needed for me to create a multilateration algorithm to track the location of BLE beacons attached to crates on a factory floor. I had nothing to go on aside from the RSSI signal strengths received by the nodes on the ceiling of the factory.

I solved this problem based on a mathematically similar problem I solved a couple of decades before, which involved calculus.

When I was still in high school, I landed my first computer job, in part... (more)

16 3 7

7 comments from Pat McCormack and more

Christopher Albertson, lived in Los Angeles
Updated October 18, 2020 · Upvoted by Bulat Ziganshin, 27 years of C++ and counting

I was a CS student. I took calculus. I can't imagine NOT knowing at least the basics of calculus.

OK, your idea of "computer science" is learning how to make a 1990's vintage web site then you don't need more than middle school math.

But today it seems AI is the big thing. How could you possibly understand how to train a network using gradient decent if you don't know what a partial derivative is? Yes you could cut and paste some code you don't understand and make it work but that is not why they train computer scientists.

Then there is robots and machine control and industrial automation that al... (more)

367 3 33

Related Ques

Why is calculus necessary in computer science degree programs?

Why does a computer science degree require calculus, but someone can learn computer...

What topics in computer science make use of calculus? What are some other sub-fields use...

Will I ever use Calc 2 or Calc 3 in computer science? If so, when?

How important is calculus in computer science?

Why does a degree in Computer Science require all the calculus courses? I doubt a...

Ask question

Messages

No messages

Connect with others on Quora by beginning a new conversation

Related Questions More Answers Below

Why is calculus necessary in computer science degree programs?

Why does a computer science degree require calculus, but someone can learn computer science on their own and get a job without ever taking calculus?

What topics in computer science make use of calculus? What are some other sub-fields used in computer science?

16 3

Answered July 21, 2018

I think that it's due to a quirk in the current state of math pedagogy. Right now calculus is a gatekeeper course - it's used as a prereq for lots of stuff, even stuff that it's not at all a prerequisite for. For instance, at a lot of schools you need to take two (or even three!) semesters of calc before you can take Linear Algebra.

Now, the way things are set up right now, there's stuff that IS necessary that's baked into the calc curriculum. For instance, working with delta-epsilon proofs in Calc 2 is going to be a lot of students' first hands-on experience with proof-based math, but there's ... (more)

3 comments from Daniel Jenkins and more

11

I can think of a few courses that would need Calculus, ***directly***. I have used bold face for the usually obligatory disciplines for a Computer Science degree, and italics for the usually optional ones.

- **Computer Graphics**/Image Processing, and here you will also need Analytic Geometry and Linear Algebra, *heavily*! If you go down this path, you may also want to study some Differential Geometry (which has multivariate Calculus as a minimum prerequisite). But you'll need Calculus here even for very basic things: try searching for "Fourier Transform" or "Wavelets", for example -- these are two very fund

... (more)

Add a comment... Add Comment

1.7K 7 16

Answered January 6, 2020 · Upvoted by Alan Mellor, Started programming 8 bit computers in 1981 and Carter McClung, B.S. UPSC General Studies & Mathematics, The University of Texas at Dallas (2006)

The first twenty years of my software career, I didn't use calculus. That carried me into the mid-2000's. I eventually found myself at a crossroads. Younger programmers (or outsourcing shops overseas) could do the same job as I was doing for much less pay. I could either move into management or find something difficult and rewarding that I could do that the recent graduates could not. I have no interest in management, so I chose the latter route.

16 comments from Abbs Nderi and more

6

Because you're getting a Computer "Science" degree, not a "programming" degree. There's a difference. While you'll most likely go on to become a programmer/analyst/developer, Computer Science is soo much more than that. In my CS program, Calculus I is required, but Calculus II can be substituted with Linear Algebra. Again, CS is a Science degree and requires you to have a certain level of math. My college also offers a Computer Science "BA" instead of a CS:BS. The math requirements stop at college algebra, doesn't require a lot of the core upper division CS theory courses, and requires a minor... (more)

Add a comment...

Add Comment

Bryant Jimin Son, Consultant, Entrepreneur, Tech Lead, Full Stack Developer

Answered July 17, 2018

Interestingly, the derivative of this question was asked more often, and it is this: "will I ever use calculus if I become a programmer" Oddly, most people say "no" when this question is posed. On the other hand, if somebody asks "why do I need to take calculus if so and so...?", I noticed more than half of people will say "you might still find it useful."