Episode Book

EPISODE #5849

Who Invented Computer Science—and Was It Ever a Trade?

PUBLISHED	10 October 2026 at 20:22 UTC
DURATION	24:54
LISTEN	myweirdprompts.com/episode/computer-science-founding-debate-85297/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.3
SOURCES	16 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:40 UTC

SYNOPSIS

Computer science was founded as a science of computation—Newell, Perlis, and Simon argued in 1967 that "the phenomena surrounding computers are varied, complex, rich." Yet the degree became the software industry's default credential, and the fight over tool users versus tool builders has run through every curriculum revision since 1968. This episode traces the field's origins: the first departments at Purdue and Stanford, the 1965 doctorates history forgot (including Sister Mary Kenneth Keller, the first woman to earn a CS Ph.D.), and what the halting problem reveals about studying a world we build ourselves. Plus: why recent CS graduates now face a higher jobless rate than philosophy majors.

TRANSCRIPT



Corn

Here's a question worth sitting with. Why is computer science the degree you get to become a software developer, when the people who invented the field would have told you that's not what it's for?



Herman

And they'd have said it firmly. That's the part people don't expect.



Corn

Daniel's asking about exactly that. The history of computer science, the visionaries who built it, how it turned from a set of ideas floating around mathematics and engineering into departments and degrees and research traditions. But the thread he keeps pulling on is the tension. The founders wanted to know what computation fundamentally is. What its limits are. What logic and information actually mean. They weren't designing a training program for a job.



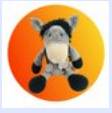
Herman

And yet here we are.



Corn

Here we are. He puts it well, actually. A physics degree isn't primarily meant to produce electricians, but computer science somehow became the standard qualification for the software industry. So how did that happen, and how does a department teach enduring principles of computation while also preparing people to work with a framework that'll be dead in four years?



Herman

And whether the modern discipline still looks anything like what its founders imagined.



Corn

That last one is the real question, I think. The other two are how we get there.



Herman

The vocational impulse was there from the beginning, though. That's the thing that complicates Daniel's framing a little. It wasn't a corruption that set in later. The first people arguing about a computer science curriculum were already arguing about whether they were training tool users or tool builders.



Corn

Tool users or tool builders. In nineteen sixty-eight.



Herman

In a panel critique of the first real curriculum, yes. And that same sentence, almost word for word in spirit, is being said in twenty twenty-six about AI. Fifty-eight years and the same argument. Which tells you the paradox isn't a mistake anyone made. It's structural.



Corn

So walk it back. Before there were departments, before there was a curriculum, who was even making the case that this was a science?



Herman

Newell, Perlis and Simon. Nineteen sixty-seven, a letter in Science, and it reads like a manifesto because that's what it was. People were asking them, is there such a thing as computer science? And their answer is one of the great opening lines in the field's literature. "There are computers. Ergo, computer science is the study of computers."



Corn

That's a shrug with a syllogism in it.



Herman

It's better than a shrug. The next line is the whole argument. "The phenomena surrounding computers are varied, complex, rich." That's the claim. Phenomena breed sciences. You don't need a natural object to have a science of something. Botany studies plants. Astronomy studies stars. Computer science studies what happens around computers.



Corn

And that was contested.



Herman

Herbert Simon had to fight this for years. There was a strong view in the hard sciences that if you're not studying a natural phenomenon, you're not doing science. You're doing engineering, or mathematics, or something in between that doesn't get a chair. Simon's answer was The Sciences of the Artificial. He argued that the artificial is its own legitimate domain, that a designed thing has its own regularities worth studying on their own terms.



Corn

Which is a bigger claim than it sounds. He's saying the made world is as worthy of a science as the found world.



Herman

Subrata Dasgupta put the distinction beautifully. If the natural scientist is concerned with the world as it is, the computer scientist obsesses with the world as she thinks it ought to be. That's the difference. An astronomer can't redesign the star. A computer scientist is building the thing they're studying, and then studying it.



Corn

That's a strange position to be in. You make the object and then you discover its properties.



Herman

And the properties were not obvious from what went into them. That's the part I find thrilling. You build a machine from logic gates, and suddenly you have a question about computational complexity that nobody could have predicted from looking at the gates.



Corn

Give me an example of that, because I think people hear "emergent properties" and nod without it landing.



Herman

The halting problem is the cleanest one. You have a machine that does exactly what you tell it, one instruction at a time, nothing mysterious in the parts. And yet you can prove, rigorously, that there is no general procedure that will tell you in advance whether any given program will finish or run forever. Not that we haven't found it. That it cannot exist. That's a fact about the made world that nobody put there on purpose.



Corn

You built the thing out of switches and discovered a wall you didn't build.



Herman

And the wall is real. It's not a limitation of our cleverness. It's a theorem. That's the moment computation stops being engineering and starts being a science with its own laws.



Corn

So the founders had their science. How does a science become a department?



Herman

Slowly, and in two places at once. Purdue gets there first. October nineteen sixty-two, they establish the first Department of Computer Sciences in the United States. And Purdue's own history of it is honest in a way institutional histories rarely are. They describe three



Herman (continued)

phases. The sixties were spent defining courses, degree programs, and indirectly the field itself.



Corn

The field itself. So they're inventing the subject as they're teaching it.



Herman

Literally. There's no textbook to assign because the textbook doesn't exist yet. The first people teaching computer science were writing the discipline in real time. Then the seventies were maturation and growth into a typical university department. And then the eighties were a series of crises, which is its own story.



Corn

What were the crises?



Herman

Funding, enrollments, the usual. But it's interesting that a field this young already had an identity crisis inside twenty years.



Corn

And the other place?



Herman

Stanford. Forsythe and Herriot start a Division of Computer Science inside the Mathematics Department in nineteen sixty-one. It becomes a standalone department in January nineteen sixty-five. So you can see the pattern. First you're a division inside something respectable, and then you cut yourself loose.



Corn

You need the math department's cover until you can stand on your own.



Herman

And that's how most of these start. Penn's is a graduate group inside electrical engineering in nineteen fifty-nine. Carnegie Mellon's proper department is July nineteen sixty-five but it grows out of an interdisciplinary program from sixty-one. Wisconsin, North Carolina, Virginia, Illinois, Cornell, Iowa, Utah, Washington University in St. Louis, all by nineteen sixty-five. It's a wave.



Corn

Ten departments in six years.



Herman

Something like that. And here's the detail I love, because it corrects a piece of folklore. Everybody says the first computer science Ph.D. was Richard Wexelblat at Penn in December nineteen sixty-five.



Corn

And it wasn't.



Herman

It wasn't, and it took until twenty thirteen for someone to properly straighten it out. Ralph London went through the records and found that two people finished on the same day, June seventh, nineteen sixty-five. Sister Mary Kenneth Keller at the University of Wisconsin, and Irving Tang at Washington University in St. Louis.



Corn

The same day. That's a coincidence you couldn't write.



Herman

Two people, two institutions, one afternoon in June. And Keller is also the first woman to earn a computer science Ph.D. Barbara Liskov at Stanford is usually cited for that, but she finished in sixty-eight. There's a good chance you've never heard Keller's name.



Corn

A nun earned the first computer science doctorate in the United States and the field's own history forgot her for fifty years.



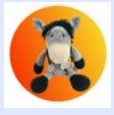
Herman

And she'd have been a strange fit for the stereotype in other ways too. She helped develop an early teaching language and argued for women in computing decades before anyone was holding panels about it. The Wikipedia-level version of this history just doesn't include her.



Corn

So the departments exist. When do they decide what the degree actually contains?



Herman

Curriculum sixty-eight. March nineteen sixty-eight, published in the Communications of the ACM. And its own retrospective says it established computer science as an academic field of study and specified to a great extent its content. That's the moment the field stops being whatever each department happens to be teaching and starts being a defined thing.



Corn

Defined by committee.



Herman

Isn't everything. And the argument they were having about the definition is the one Daniel's pointing at. The critique of that curriculum, at the time, asked whether computer science programs should prepare their graduates to be tool users or tool builders.



Corn

Which is a beautiful way to put the two jobs.



Herman

And look at the persistent issues. Across all four sets of curriculum guidelines, sixty-eight, seventy-eight, ninety-one, two thousand one, the same two fights come up over and over. What role does mathematics play, and how much programming belongs in the introductory courses.



Corn

Fifty years, four revisions, same two arguments.



Herman

Which is either reassuring or damning depending on your mood.



Corn

I find it reassuring. It means those weren't questions anyone got wrong. They're questions that don't have a settled answer. You renegotiate them every generation because the terms change.



Herman

That's the thing about the founders' ambitions, actually. They wanted a science of computation, and what they got was a department that has to teach a trade, and has to keep re-deciding the ratio between the two. And it's worth saying the vocational side was present at the founding. The ACM's preliminary recommendations in sixty-five explicitly addressed instruction in the use of computers as a tool as a separate need from



Herman (continued)

computer science proper.



Corn

So a student in sixty-five could take a course on using a computer, and that was understood to be different from studying computation.



Herman

Understood by everyone at the time. Which means the vocational impulse didn't creep in. It was in the room the whole time, sharing a table with the scientific one, and nobody could agree which one was the guest.



Corn

Here's what I find strange about that, Herman. Those two functions got absorbed into a single degree. There was no separate track. Nobody built a vocational program with its own name.



Herman

I went looking for exactly that and there isn't one. There's no distinct computer science as vocational training degree that ever got established. What happened is the standard major swallowed both jobs.



Corn

The science major became the industry credential by default, not by design.



Herman

By default. And that's the accident at the center of this whole thing. Because the alternative, a trade school qualification, would have been perfectly respectable. We have those for plenty of professions. What we got instead is a degree whose stated purpose is to teach you what computation is, being used as the filter for whether you can be trusted to write a login page.



Corn

A login page.



Herman

I've seen the job postings.



Corn

So when does that start to hurt?



Herman

When the credential stops delivering. And there's fresh data on this, from the last couple of weeks. The jobless rate for recent computer science and computer engineering graduates is around seven point one percent.



Corn

Seven point one.



Herman

Which by itself doesn't sound catastrophic. Here's what makes it interesting. That's higher than the jobless rate for philosophy majors, art history majors, ethnic studies majors. The degree that was the sure thing is now underperforming the degrees people joke about.



Corn

That's a number that would have been a punchline ten years ago.



Herman

And the supply side explains a lot of it. Recent computer science and computer engineering graduates tripled over a decade, to about three hundred and sixty-two thousand in twenty twenty-four. Then enrollment started falling. Down eight point four percent at four-year institutions by spring of twenty twenty-six.



Corn

So the curve turned before most people noticed.



Herman

And Michigan is the case study. Their engineering computer science declarations went from seven hundred and forty-seven in twenty twenty-three to six hundred and five, then three hundred and ninety-nine, then two hundred and ninety-four this year.



Corn

That's a sixty percent drop in three years.



Herman

Their liberal arts CS numbers are the same shape, seven hundred and twenty-eight down to two hundred and seventy-four. And in September, a couple of weeks ago, Michigan ended the special admissions process for the major entirely. Opened it to any student who completes the prerequisites, because demand stabilized.



Corn

Stabilized. That's the word they chose.



Herman

Every department that's suddenly admitting everyone calls it stabilizing.



Corn

It's a nice bit of institutional vocabulary. What's the actual mechanism though. Is this AI eating the entry-level jobs, or is it just the ordinary cycle of a hot field cooling?



Herman

Both, and the evidence for the AI part is more specific than I expected. Stanford's Digital Economy Lab looked at occupations they class as AI-exposed, software development being the obvious one, and found employment for workers in their early twenties is nineteen percent below where it would be if those occupations had tracked less-exposed fields.



Corn

Nineteen percent below trend.



Herman

Below the counterfactual. And a separate Stanford study found entry-level software development employment for ages twenty-two to twenty-five fell nearly twenty percent since twenty twenty-four.



Corn

So it's not that software jobs vanished. It's that the bottom rung of the ladder got pulled up.



Herman

That's exactly how Tracy Camp at the Computing Research Association describes the danger. If they don't change how hiring is happening, they're not going to have mid-level career people in a few years. You're not replacing juniors with AI. You're deferring the people who'd become your seniors.



Corn

Which is the kind of thing that looks fine for three years and then looks like a hole in the org chart.



Herman

And the employers are asking for something strange. Nicholas Mattei at Tulane put it well. Companies come back and say you need to train these people even better, and they need to be able to jump into a middle management role as opposed to an introductory or junior developer role.



Corn

They want a graduate with no experience who can run a team.



Herman

And Mattei's own verdict is that it's kind of crazy. Which it is. You've removed the apprenticeship and you're demanding the product of the apprenticeship as a hiring condition.



Corn

There's a piece of this I keep circling back to, though. The department chairs aren't panicking.



Herman

Atul Prakash at Michigan specifically isn't. His line is that long term, computing is not going anywhere, and his expectation is there will still be plenty of jobs in computing fields long term. And he compares the current decline to the dot-com bust.



Corn

Which is a reasonable comparison and also the thing everyone says during every downturn.



Herman

Fair. But he has one advantage over the people saying it in two thousand and one, which is that he's seen how that one ended.



Corn

And the students?



Herman

One student at Michigan, Kavya Aggarwal, put it about as plainly as it can be put. Your CS major just doesn't matter as much anymore.



Corn

That's a nineteen-year-old saying what it took the rest of us a year of job listings to figure out.



Herman

And it lands differently coming from someone who's paying the tuition. The credential has stopped doing the work it used to do, so it's reverting to being what it always was underneath. A degree in a subject you either find interesting or you don't.



Corn

Which is arguably the healthy version. But I want to go back to Daniel's real question, because I think the two halves of it are in tension in a way he might not have intended.



Herman

Go on.



Corn

He asks whether the modern discipline still reflects the original ambitions. But the original ambitions were never purely scientific, and that's the thing we've established. The founders wanted a science of phenomena. The ACM wanted tool users trained. Those two groups were in the same room in nineteen sixty-five and they've been in the same room ever since. So the question isn't whether the field drifted. The field was born already split.



Herman

And the drift he's sensing is really the balance tipping. For most of the field's life, the vocational side could be served by teaching the science well. Learn how a compiler works and you can write one. The theory paid for itself.



Corn

And now?



Herman

And now the tools change faster than a curriculum cycle. Four years of coursework and the framework you learned in your first year is two generations old. The theory doesn't expire but the practical half of the degree does. So departments are stuck teaching the enduring part well and the perishable part constantly.



Corn

Which is the actual structural problem. Not that they've betrayed the founders. That they have to service a client, the employer, who is not the same as the student, and whose demands change every three years.



Herman

And the four curriculum revisions are just the field admitting that. Sixty-eight, seventy-eight, ninety-one, two thousand one. Four times they sat down and renegotiated what a computer science graduate is. The honest reading of that record is that there is no stable answer, only a rolling argument.



Corn

Here's a thing I find funny about it, though. Daniel's framing is that computer science is the obvious degree for a software developer, and that's the anomaly. But there's a parallel that gets overlooked. Medicine isn't a science degree either.



Herman

Explain that one.



Corn

A medical degree is vocational. Explicitly. It trains you to be a doctor, and it makes no apology for it. Nobody sits around wondering whether the medical curriculum has betrayed the original ambitions of physiology. The vocational purpose is stated on the front of the building.



Herman

So the anomaly isn't that CS has a vocational function. It's that CS has a vocational function it never admitted to having.



Corn

Which is a much more interesting claim, and I think it's closer to right. The department that calls itself a science and quietly functions as a trade school is a specific English-speaking arrangement. It's a bit of a strange one.



Herman

There's one more piece of this I want to put on the table before we move on, because it's the counter-signal to all the doom. Ken Griffin gave Carnegie Mellon three billion dollars.



Corn

Three billion.



Herman

On the thirtieth of September, to build a Miami campus. Three billion dollars, the week after we're told computer science graduates can't find work.



Corn

Somebody is planning on the field being fine.



Herman

Somebody with a lot of money and no particular reason to be sentimental about it. Which is the thing I'd hold onto alongside the enrollment numbers. The enrollment is falling. The philanthropy is not.



Corn

Hm. Although I'd want to know what the three billion is actually for. A campus is a real estate bet as much as an academic one.



Herman

Fair, and I don't know how they've structured it. But it's a bet.



Corn

It is. And the reason it isn't crazy is that all the pain we've been describing is concentrated at one end. The entry-level end. The research end, the graduate end, that's where the three billion goes.



Herman

And that maps onto the founders' problem, in a way. Their science was never the part that was struggling. What's struggling is the part the founders didn't quite want to own. The trade.



Corn

Which leaves us with a question, or at least with me. If you're eighteen and you're choosing a degree, and the credential no longer guarantees the job, what's the actual argument for the degree?



Herman

I have an answer, but I don't know if it's a satisfying one. The argument for it is the same as it ever was. It's an interesting subject.



Corn

Which is a very Hannah answer.



Herman

Who?



Corn

Nobody. Forget it. What I mean is that's the answer you give when the second reason has collapsed and you're left holding the first.



Herman

Then maybe the honest thing is that the field is being asked to be two things and the market is only paying for one of them this year.



Corn

And the one it's paying for is the one the founders would have called the lesser thing. That's the irony Daniel should sit with. Not that the field sold out. That the market finally picked, and it picked the trade.



Herman

For now.



Corn

That's the part nobody can call.



Hilbert

— and the man had three of them stacked in the back of a station wagon, which is why I say you can't trust the number.



Corn

Sorry, three of what?



Hilbert

The teletypes. Spring of eighty-three, maybe eighty-four. I was doing the audio for the department's colloquium series at the time, and the university had a deal with a supplier who loaded the equipment into a wagon rather than shipping it, because the freight in that era would have eaten the whole discount.



Herman

That's the sort of arrangement that exists for a reason.



Hilbert

It exists because the purchasing office wouldn't sign off on anything under a certain weight. So they'd drive it, and the drivers were the suppliers themselves. This is a detail nobody has, but you were saying the two first doctorates were the same day. That's true. June seventh, sixty-five. But the defense wasn't the hard part for the Wisconsin one.



Corn

What was.



Hilbert

The building. The air conditioning in the room they'd booked broke that morning. The chair wanted to move the whole thing to a smaller room, and there was a real question of whether they'd have to postpone. They ended up holding it where it was with the windows open, and there was a fan on the floor pointed at the candidate.



Herman

You've told that before.



Hilbert

I've never told it before. The janitor on that floor was my uncle. He was the one who wheeled the fan in, and he's the one who told me, which is why the version you two have is missing the fan.



Corn

Your uncle was a janitor at Wisconsin.



Hilbert

He was a janitor at Wisconsin. He was also doing the overnight shift at a radio station in Milwaukee four nights a week. Herman, I heard your audio levels drop when you got excited about the Stanford numbers, by the way. Bring the mike closer or it swallows the first consonant.



Herman

Noted.



Hilbert

Where was I. The radio. He did the overnight show four nights a week, which is why he was never on the floor before noon. He interviewed a young Alan Kay once, so he claimed. I never got the tape. And no, before either of you says the obvious thing, I've done the geography. Milwaukee to Madison, four nights a week, in the middle of the night, on the roads they had then.



Corn

Which is not physically possible.



Hilbert

I didn't say it was possible. I said he did it. There's a photograph of him with the nun at a church picnic in seventy-two, and every time I look at it I'm more sure it's her, except that it isn't in a church and she isn't wearing the habit, so it's really a guess.



Herman

You're sure it's her, based on a picnic.



Hilbert

I'm pretty sure. And that's about as far as I go with it.



Corn

Hilbert, why is the fan the detail you've been saving.



Hilbert

Because the story everyone tells is the one about the doors opening for her, and nobody tells the part where the room was too hot and a janitor found a fan. That's how a thing that's going to look historic actually happens. Somebody solves the small problem. Now, the levels, Herman.



Herman

Which leads us to the part I keep coming back to. If Hilbert's right that the actual barriers were never the grand ones, that changes what the enrollment numbers mean.



Corn

They mean the field has to make its case on the science again, because the credential argument has stopped working. And that might be the healthiest thing that's happened to it in fifty years.



Herman

Whether that's a cycle or a permanent reset is open. Prakash says cycle, the Stanford numbers look less like a cycle, and I don't think anyone can call it yet.



Corn

And the field has been here before, in a sense. Four curriculum revisions, each one renegotiating what a computer science graduate is. It survives by redefining its core.



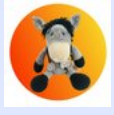
Herman

The open question is whether it can redefine it one more time without drifting so far toward the trade that the science stops being the point.



Corn

That's a question Daniel can sit with.



Herman

We should credit our producer, Hilbert Flumingtop, who has been here the entire time and who apparently has opinions about fan placement.



Corn

If this was your kind of episode, go back for episode thirteen, AI; episode twelve, The AI Breakthrough; and episode eighteen, Beyond the GPU. This has been My Weird Prompts.



Herman

If you've got a question of your own, send us your own prompt on Telegram at t dot me slash MWP listener bot.



Corn

We'll be back soon.



Herman

See you then.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **What is Computer Science [primary]**
   Newell, Perlis & Simon, *Science* Vol. 157, Issue 3795, pp. 1373, 1374, 1967-09-22
   https://iiif.library.cmu.edu/file/Newell_box00048_fld04157_doc0005/Newell_box00048_fld04157_doc0005.pdf

2. **Purdue Department of Computer Science history [primary]**
   https://cs.purdue.edu/history/

3. **Purdue CS milestones table [primary]**
   https://www.cs.purdue.edu/history/table1.html

4. **Stanford CS Department timeline [primary]**
   https://wgetsnaps.github.io/cs.stanford.edu--calendar-2016-01/about/department-timeline.html

5. **Curriculum 68 [primary]**
   ACM Curriculum Committee, *CACM* Vol. 11, No. 3, March 1968
   https://dl.acm.org/doi/10.1145/362929.362976

6. **History of computer science**
   Wikipedia
   https://en.wikipedia.org/wiki/History_of_computer_science

7. **Who Earned First Computer Science Ph.D.?**
   Ralph L. London, Communications of the ACM, 2013-01-15
   https://cacm.acm.org/blogcacm/who-earned-first-computer-science-ph-d/

8. **The genesis of computer science**
   Subrata Dasgupta, OUPblog, 2014-02-07
   https://blog.oup.com/2014/02/genesis-computer-science-dasgupta/

9. **The Science in Computer Science**
   Communications of the ACM
   https://cacm.acm.org/opinion/the-science-in-computer-science/

10. **Computer science as empirical inquiry: symbols and search**
    Newell & Simon, *CACM* Vol. 19, No. 3, 1976-03-01
    https://dl.acm.org/doi/10.1145/360018.360022

11. **Department of Computer Science (Purdue University)**
    Wikipedia, (Purdue_University)
    https://en.wikipedia.org/wiki/Department_of_Computer_Science

12. **The ACM and IEEE-CS Guidelines For Undergraduate CS Education**
    Communications of the ACM
    https://cacm.acm.org/news/the-acm-and-ieee-cs-guidelines-for-undergraduate-cs-education/

13. **As the coding boom fades, computer science grads focus on AI skills in choppy job market**
    AP via Japan Today, 2026-09-27
    https://japantoday.com/category/tech/as-the-coding-boom-fades-computer-science-grads-focus-on-ai-skills-in-choppy-job-market

14. **UMich computer science no longer requires specialized application as demand steadies**
    Michigan Daily, 2026-09-29
    https://www.michigandaily.com/news/academics/umich-ends-computer-science-specialized-admissions/

15. **Ask HN: High school student**
    is learning programming still worthwhile?, Hacker News, 2026-06-04
    https://news.ycombinator.com/item?id=48403614

16. **Hedge fund CEO Ken Griffin gives record $3B to Carnegie Mellon**
    Chicago Tribune, 2026-09-30
    https://www.chicagotribune.com/2026/09/30/ken-griffin-carnegie-mellon/