

# Will ChatGPT replace programmers?



## Tycho Henzen

Software Engineer(C++) at VRee Experiences (2016–present)[Sun](#)

No.

For multiple reasons:

1. ChatGPT, or any similar system, breaks down beyond toy problems, it's memory cannot simultaneously accommodate the thousands of "truths" present in any project bigger than a handful of classes, chatGPT's memory capacity is around 3000 tokens, so it can't effectively reason about a 10,000 LoC codebase
2. ChatGPT is not capable of independent thought, it requires a prompt explaining the desired end result, which is effectively the design phase of programming. Actually writing the code is just a translation step from that conceptual goal.
3. ChatGPT does not consider the broader context of your application beyond what you've explicitly told it
4. As a consequence of 1 - 3, you'll need developers to design good prompts if you hope to effectively use it for dev work, though it may lower the barrier to entry slightly.
5. However, Programming is the one profession where automation just creates more demand, if chatGPT makes 3x as many developers work 10x faster, we'll just think of 50x more projects to do.
6. ChatGPT is dumb, it just acts smart, it actually can't even do basic arithmetic so any question asked has a 25% chance of being answered dead wrong, and if you're not smarter than chatGPT on the topic of the question, you just won't spot the mistake



who was older when they were elected president: grover cleveland or george bush



Grover Cleveland was the older of the two presidents when he was elected. Cleveland was born on March 18, 1837 and was elected president for the first time in 1884, at the age of 47. George H. W. Bush was born on June 12, 1924 and was elected president in 1988, at the age of 64.



is 47 larger than 64?



Yes, 47 is larger than 64. In general, a number is considered larger if it is greater than another number. In this case, 47 is greater than 64, so it is considered the larger number.

587 views

View 4 upvotes

Upvote

4

Sponsored by SonarQube

### Static code analysis in your language of choice.

Detect bugs, review security hotspots, track code smells, & fix technical debt. Start free.

Download

Upvote

14



### Håkon Hapnes Strand

.

[Follow](#)

CTO and Machine Learning Engineer [Dec 9](#)

ChatGPT isn't even made with programming as its primary use case. It's optimized for conversation. It just happens to be pretty good at programming because its base model is trained on a huge corpus that also includes code.

OpenAI does have a model that *is* optimized for writing code that is called Codex. However, that one is built on the iteration of GPT-3 that was available last year.

The new and updated GPT-3.5, which has been finetuned with reinforcement learning, is so much better than last year's model that its derivate models, ChatGPT and the sibling

InstructGPT model called davinci-003, are probably better at writing code than Codex itself at the moment.

However, there will surely be new iterations of Codex that are even better. And it's likely that Google will launch a competing product.

In the not-too-distant future we will see programming tools built on top of these models that will make the current iteration of ChatGPT look primitive in comparison. At that point, you can probably start to replace mediocre web developers with AI. And in 10–20 years... who knows?

The rate of improvement in AI is accelerating, and it's almost impossible to predict what will be possible just a few years into the future.

11K views



## Tim Mensch

Freelance CTO/Software Architect [Updated Dec 14](#)

No.

Not unless users are suddenly going to be satisfied with short command line programs written to do some tiny part of what they actually need.

And that half the time don't even work, or at least don't work the way you'd need them to.

It's a cool magic trick. But that's all it is.

Well, *maybe* it can be useful enough to be a coding *tool that helps programmers*. But note that means we still need the programmers. They just won't have to type as much.

Thing is, the "hard" (or rather, *skilled*) part of programming is not, nor has it ever been, the typing.

Maybe some less-skilled developer teams will need fewer developers to achieve the same productivity. But if you're talking about less-skilled developers, they also have been in danger of losing their jobs to outsourcing. To actual humans who often cost less than \$20/hour.

And who can do more than write short command line tools.

Keep in mind that non-programmers often don't even know what they actually need. I mean, they *might* know how the end result should work, though it is usually best to get a UI/UX expert involved. But even a perfect description of how an app should work could

have dozens of different possible interpretations by an AI that fundamentally doesn't understand what it's doing.

And non-programmers often really, really don't want to dig in the weeds and build their own product. Just look at how many drag-and-drop site builders exist, and yet there is still a huge market for developers to assemble WordPress sites. I would argue that using a drag-and-drop no-code site or app builder is easier *and* more reliable than using ChatGPT.

And Wix, the drag-and-drop site builder, has been around for **16 years** and it still has yet to replace programmers who do its exact job. (Yes, programmers can build sites impossible to create in Wix, but many sites would be fine, cheaper, and some even better, just using a site generator like that.)

So much angst about the death of software jobs, and the answer is always the same: As long as there are jobs for *anyone*, there will be jobs for skilled software engineers.

And I suspect that even mediocre software engineers will keep their jobs for the most part, just due to the way that large companies work: Prestige of managers and executives is often proportional to the size of your team or department. So they're actively motivated to not fire people unless they have no choice—and that won't change just because a new tool makes their team more productive.

Heck, it's more likely that they'd simply come up with more things for their team to do.

---

Please stop stressing over AI taking over your job. It really won't and can't, and your boss doesn't probably want to fire you even if it could.

6.2K views

View 99 upvotes

View 4 shares

Answer requested by

[William Parrish](#)

[Will ChatGPT replace Google?](#)

[Have you tried out ChatGPT? Is it better than Google Search?](#)

[What is the future of web development after ChatGPT? Will programmers lose their jobs?](#)

## Ryan Hansen

Software Engineer (1998–present)[Sat](#)

AI constructs may eventually alleviate software engineers from needing to write code themselves but it will never replace the need for engineers in the first place. Fingers hitting a keyboard is a very small fraction of what I do for a living. It is easily the most tedious and monotonous aspect of my job. It is the functional equivalent of swinging a hammer or digging a ditch.

If it were capable of replacing me then there isn't a single job on the face of this planet that it wouldn't be able to replace. There wouldn't be roles left for humans in fields like entrepreneurship or even marketing. Engineering literally sits at the intersection of scientific understanding and artistic expression. A non-insignificant element of the value that I add to an organization comes from the exact same capacity for imagination as any other human endeavor.

Think of AI as a tool that humans can use to make their lives easier. An AI simply means I spend less time with a keyboard just like a nail gun means a carpenter will spend less time swinging a hammer.

572 views

[View 3 upvotes](#)



## Dave Voorhis

.

[Follow](#)

Software entrepreneur, engineer, and educator for 35+ years.[Upvoted by](#)

[Alan Mellor](#)

Started programming 8 bit computers in 1981 and

[Marcelo De Zen](#)

, 15+ years as a software developer.[Dec 10](#)

It was brought up on our work Slack yesterday, and having the day off but stuck at home whilst the roof was being put on our new porch, I spent a couple of hours playing with it.

I made numerous efforts to get it to produce code I could personally use, and not once did it produce anything that was remotely usable, at least not with more work than creating it from scratch myself. At best, it produced examples of the sort I once provided

in university classes to give a good idea of what an assignment required without giving away the real solution.

Where it might have use is generating regular expressions. I asked for several standard examples plus some custom problems, and it generated results. But I'm not sure they were better, or that it wasn't more work, than using a conventional regex generation tool.

I'm also not sure I'd trust it to be correct.

I asked it to give me the last digit of pi. It confidently, though with some reasonable caveats, claimed it was 9, then gave me an example of pi with a dozen or so digits to demonstrate this fact. They ended in a 6.

I asked it for several fictional stories on given topics, then several interviews with famous historical figures. The results were all notionally competently written — the grammar was correct and they were easy to read — but dull and superficial, like those written by a bored high school student or a clickbait content writer who knew what the topics were about or what the historical figure did, but didn't care.

I asked it for instructions for starting a vintage motorcycle. The response was accurate, mostly, except for entirely leaving out the important bits that would avoid the risks of injury from misusing the kick-starter.

A colleague did a similar request for instructions on how to land a glider. He said he thought it was trying to kill him.

Attempts at natural conversation seemed stilted, dull, and joyless. It felt like a bad dinner date with a slightly stern, humourless and judgemental person who didn't like me.

Speaking of dinner, it did surprisingly well at generating recipes. Requests for vegan stew with chicken were appropriately rejected as contradictory, but (say) a request for vegan stew with lentils and potatoes produced what looked like a recipe for a pleasant, though perhaps somewhat uninspired dish. I suspect it would be a bit bland.

Though I'd happily use it for that.

My overall impression was that it's what we should expect from a modern-era ELIZA, but like ELIZA it's still basically a toy, though it might make a great core for something like Google Assistant or Amazon Alexa.

But replacing programmers?

No, not by a long shot. When it can interview users and stakeholders and design a solution in light of company direction and corporate vision, whilst considering budget, timescale, available and preferred infrastructure, industry regulations and current technology trends and meet functional and non-functional requirements, then I'll be concerned.

For now, it doesn't even look like it could be a useful programmer's assistant; a capable mechanical co-worker that can quickly hammer out the repetitive dull parts so I can focus on the good bits.

I think we'll have to wait for GPT4, or GPT5, or later, before we get that.

315.2K views

View 517 upvotes

View 7 shares

Upvote

517

84

7

Sponsored by nexo.io



**Peter Shaw**

[Follow](#)

I've been writing software for over 30 years, on many different platforms. Upvoted by

[Alan Mellor](#)

Started programming 8 bit computers in 1981 [Dec 10](#)

NO ... NOT A CHANCE

---

BUT

There will be many who will tell you otherwise.

WHY?

---

Simply because they know, no better.

Lets take a very good example, that Tim Corey used to illustrate the problems we fact with this new technology.

He asked the ChatGPT AI to create a piece of code that generates random numbers between a lower bound of 5 and an upper bound of 10, and at first glance what the system generated was perfectly fine and working code.

It generated the following snippet:

```
1. public static int GenerateRandomNumber(int bottom, int top)
2. {
3.     Random rnd = new Random();
4.     return rnd.Next(bottom, top);
5. }
```

But to anyone who's been in the industry for a while, there are a number of problems with this code.

Lets start with the "Concurrent Access Problem", see how every time you use the function, it generates a NEW random object, using the .NET random assemblies provided for you?

Well, when you call a new on Random, like that, with NO SEED value, it uses the system timers to seed the random generator.

Now imagine if this is called from an application running multiple threads and processes, and 2 calls come in from application code to generate random numbers, each within about 1 or 2 micro seconds of each other.

BOTH calls are going to generate the same number, there will be nothing random about them at all.

Now that might not matter to you, but imagine if this is being used to generate something simple like a salt value for blowfish encryption in a database. You might suddenly find that someone's hashed password, now has exactly the same salting value as a different person.

Then there's the fact that it's declared static.

If we imagine, that your going to go on to use this code to add a static generation method, into a larger sealed class, if you don't know any better, then your going to add something into a class object, that can be called without newing up an instance of an object.

Going back to our random number for a salt value previously, that might mean that an attacker, now has a way of setting and KNOWING a salt value, before a random number is generated, or they may be able to influence the salt value, making it something they then know later, which might make it easier to brute force attack the password hash values in a database.

There are other problems too, but my point here is NOT to point out the failings of ChatGPT, my point is that the code it generates CANNOT BE TRUSTED.



---

Beginner developers especially, and those that are in-experienced, DON'T REALISE that correct code is NOT always correct.

Take this very simple 2 line statement:

```
1. var name = CustomerObject.Name;
2. Console.WriteLine("The Customers Name is " + name + " if you
   wish to contact them");
```

To someone who doesn't know better, that looks like perfectly valid code. And it is...

It will compile correctly, it will run correctly, as long as "CustomerObject" is defined.

In C# if CustomerObject is not defined, or has at the point of use, not been newed up and assigned some sensible values, it will generate a null reference example and crash the application.

Now take the following 2 line statement (Which does exactly the same thing):

```
1. var name = CustomerObject?.Name
2. Console.WriteLine($"The Customers name is {name} if you wish
   to contact them");
```

Looks basically the same right?

but notice the question mark after the object reference, and the different way of interpolating the data item in the string.

That last example, will execute without causing any problem, other than the name just will not get printed.

It won't throw an exception or crash the application if the value does not hold a meaningful value.

Something like ChatGPT that is only responding to input that is as knowledgeable as the person inputting it, cannot know these subtle differences.

---

OK, so you might now be thinking to yourself, well I'll just be really specific and ask ChatGPT to make sure it doesn't generate faults in the code like that. After all, it's a very clever technology, and it should understand me if I ask it specifically what I want right?

NOPE ....

Tim shows quite clearly in his video on the subject, that while you can be very specific about what you want, ChatGPT does not remember everything.

In one case where he asks it to generate some database code using Dapper, he has to repeat the question twice, and remind the system he asked for dapper previously.

This "specificity" and verbosity of the description you have to ask for a complex piece of code, could easily end up being way longer and more complex than any code you might have to write manually.

---

If software developers get to the point where they feel they don't need to learn, and can just ask a system like ChatGPT for an answer, and then they just take the answer they are given on blind faith without any validation as to it's worthiness for the task at hand, and they are going to start causing a lot of problems when they do.

---

Already I'm seeing business folk looking at this through the eyes of "How many software developers can we remove from our payroll", or "How can we use this to write profit generating apps for us, without paying for staff"

Already I'm seeing people who "want to be developers" but don't "Want to put in the work required to train and become on", eyeing this technology up and thinking "Oooooo, how rich could that make me"

And Stack Overflow the programmer Q&A site has already posted a statement saying they will remove any answers that appear to have been generated by ChatGPT, along with banning those users making use of it.

---

People ARE GOING TO ABUSE IT, people ARE GOING TO DEPEND ON IT as a source of truth and people ARE GOING TO IGNORE THE WARNINGS in favour of the money they believe they can make from it, but that attitude is going to end in disaster, unless you take the code generated by it for what it is.

Code that is generated by cross referencing 1000's of blog posts just on that one subject, then generating a code snippet based on what the average general response to the question is.

---

ChatGPT is a powerful tool, there is NO DOUBT about that, but to trust it 100% implicitly to generate a bit of code you need is going to be very foolish.

It knows NOTHING about your project, It knows NOTHING about your required features, or what the client wants, or how the project treats security, or what hardware the application will run on, and the many 10's of 1000's of other variables involved in deciding how a bit of code should be written.

Context counts for more than you think in ANY software project, that a developer works on, and no system "ChatGPT" or otherwise can ever hope to figure that side of the equation out.



If you want to watch Tim Coreys video on the subject, it can be found here:

1.5K views

View 16 upvotes

Answer requested by

[William Parrish](#)

Upvote

16

7

Sponsored by Aspose

### **Create, read, convert and process GIS data programmatically.**

Aspose GIS enables you to manipulate geographic information from vector-based geospatial data formats.

[Learn More](#)

Upvote

1



[Alan Mellor](#)

.

[Follow](#)

Experienced in C, C++, Java and GoUpvoted by

[Marcelo De Zen](#)

, 15+ years as a software developer.[Dec 12](#)

- It is not intended to do that

- If you choose to do that, you will have a lot of badly broken code mixed in with working code
- Like an annoying coworker, it delivers highly confident incorrect explanations about why its broken code is perfect. They sound very convincing. "Wanna buy a timeshare?" says GPT
- Our industry has managers who cannot tell working code from defective code. This does not bode well for a team replaced by ChatGPT in its current form.

Should it? No.

Can it? No.

Will it? Sadly, programmers will have no say in this matter, once again. It might.

Upvote

257

29

6

Sponsored by Aspose

### **Create, read, convert and process GIS data programmatically.**

Aspose GIS enables you to manipulate geographic information from vector-based geospatial data formats.

Learn More

Upvote

1



### **Paulina Jonušaitė**

.

[Follow](#)

has extensively studied and applied different languages for more than 15 years. Upvoted by

### **Alan Mellor**

, Started programming 8 bit computers in 1981 [Dec 11](#)

ChatGPT so far has proven one thing – it can generate text and code that **seems** competent to people who don't know what they are looking at. If you need to convince someone clueless like Elon Musk that you have been doing your job, ChatGPT is irreplaceable.

For everyone else, ChatGPT at best produces first-draft level material. A lot of it is wrong, most of it can be salvaged, very few things are up to snuff on the first try. There are contexts where this is useful – reviewing content is easier than generating it – but it is still nowhere near replacing an *intelligent* human.

It could completely demolish sports journalism industry though.



## Charles Guerin

### Follow

10 years running aerospace company, 35 software architect. [11h](#)

It will require a new type of programmer more akin to an analyst. Let me explain.

I've worked with Github CoPilot, ChatGPT and the OpenAI API, and used the latter to train models to solve quality control issues in my company.

All these tools are based on the same basic AI model: GPT-3

Amusingly, I did an experiment where I had CoPilot write code to access OpenAI's coding model and teach it to write console applications, compile them (using dotnet) and then execute them based on an english text.

That was interesting because I have it setup so that I can explain a console application in English (or French seems to work too), and it will generate a compiled, running application.

Other people answering this have mentioned the 3000 token limits to its memory, and that's true, but it doesn't take too much imagination to figure out a way around that. Its a matter of how you partition code and a general rule of coding is that any class should only have one purpose. So by layering your requests, and building a few tools around this stuff, you could conceivably build a very large project. Add to that that AI is really useful for unit testing code, and it's just a matter of time and someone developing the right structure.

The real answer to your question in my mind, as I'm working on more "meta" uses of the tools is that a new type of Software Development skill that's more akin to Software Analysis than development is required.

I've found that when you try to get these tools to create something you have to VERY VERY specific about the rules, so the person with the skill who knows EXACTLY WHAT THEY WANT to see in a program is going to have the easiest time creating software with GPT-based coding.

As an aside — I found that OpenAI Davinci model is the best at straight coding from a clear requirement. CoPilot comes in second place. ChatGPT makes MANY errors, however when you point out the errors, its quite apologetic (which amuses the hell out of me) and it gets it right the 2nd or 3rd time. So, that one is not ready to replace any programmer.

I do enjoy the ChatGPT model for development though. Since I'm a very senior developer, I can spot the problems quickly and I appreciate the way that ChatGPT constantly explains it's logic, even when the logic is a bit goofy. I think for current developers, a workflow between the tools is the way to go:

- 1) Use ChatGPT to discuss requirements and programming ideas and explain them to get an idea for what they might look like and generate class "stubs".
- 2) Use OpenAI API to automatically generate code like SQL code and procedures, tables, Rest interfaces automatically by writing a few tools to autogenerate the files you might need.
- 3) Use Github CoPilot to help create inner code inside the "stubs", along with a sane developer monitoring for code quality.
- 4) Use either CoPilot or ChatGPT to write unit tests based on requirements, the same way we use languages like "pickle" to

I've found OpenAI is best at creating file convertors, and teaching it complex JSON/XML structures is a breeze.

One amusing story — we have a customer that often sends us relatively complex human-created XML files with formatting errors. We taught the AI to spot the errors and wanted to create an app to point out where the errors occurred so that the customer would have a tool to see the errors. The problem developing the tool was that the AI had such a good knowledge of the file format and the customer errors after training that it kept fixing the errors instead of pointing them out.

So ultimately, I decided to do both, provide one mode that pointed out the error and then simply output the fix... a very unusual thing from my POV.

57 views

View 1 upvote

Upvote

1



**Brandon Ross**

[Follow](#)

Will ChatGPT replace programmers?

No?

ChatGPT is a basically response generator. Read my brief summary here:



Brandon Ross

· Dec 13

### **Should we be scared of ChatGPT?**

No. It'll be a great tool. Very good. But it's easy to forget what it is and how it works. And so, what it is not. Effectively—and with much handwaving—it's a super SUPER big "next word" generator. Your predictive text keyboard wearing an infinity gauntlet. So hugely big, it can generate seemingly sensible text in response to a very wide range of requests. But it does this through a very large probabilistic model. Some philosophical boundaries aren't far away, but it's very hard to argue that ChatGPT comprehends what you're asking. For example: How do you cut a tomato? What would the first word be? If you saw enough questions like this: "How do you x a y?" y "To x a y" Now, to ChatGPT, the string "How do you" isn't even necessarily three different words—it's just a string of text. It's effectively: "[REQUEST EXPLANATION] x a y." You might notice that most questions like this, get answers in a form like: "To x a y you first take a y and" What's next? Find whether y is a discrete object... So now, find other nouns associated with x'ing the y. "To cut a tomato you first take a knife and then" And you just keep applying a very large set of rules, trying to predict what words come next. To be very clear, I wrote the above explanation above before I asked ChatGPT the above question. ChatGPT shows that if you have enough data to feed into this system, the relations and associations for many contexts can be discovered. And it has some limited "working memory" of its requests and its responses. But it's not really doing much more. SkyNet it ain't. So, the risk here is that a human can ask a seemingly complex question, and mistake this depth of training for more abstract conceptualization and understanding. This is easy to do, because most people rarely think too deeply about what their bandwidth limited-words mean. That is, as actually communicated and separate from their ideas about those symbols, occurring in their minds. It's a common source of everyday miscommunication. Though it's rarely thought about that way. And the range of ordinary communication leaves many answers sound as if they have intentionality behind them. The reality of much human communication, is that it IS rather thin. It often doesn't really matter whether you understand what you're saying. But, within 60 seconds, I was getting stuff like this: \* More like OpenDERP. \* Here, should be "Hello" \* But it could be anything! Rather, the text hasn't been created yet. There is no fixed answer. \* Similar. \* I don't do X, but I do X. \* Because it doesn't really understand or verb abstract nouns. \* Sounds like Wikipedia was part of the training data. \* Note the almost formulaic answer. \* It's seen this trick question a few times. It associates it with something entirely different. \* Total fail. So for

what it is, it's neat. It will make for some really interesting and slick interfaces. And some generative applications we probably don't even know we want yet. (And it means that customer service is about to get even more unhelpful.) But... It's also kind of deeply unfulfilling on the research front. A bigger and better ML model is nice. But it isn't a very satisfying advancement. And I think it's misleading to strongly call it AI.

ChatGPT has been trained on a large body of written materials. And that includes a lot of written materials about various programming languages.

So, if you ask ChatGPT questions about certain programming tasks, you will get some programming-like output. Your question is associated with that kind of material.

That doesn't mean the output is going to be sensible.

As the example above make clear, it doesn't mean it's going to be what you *want*. Or that it will make logical sense.

But if you need a code snippet to do something very simple, that information *might* be encoded within ChatGPT's language model. (While programming information might seem more "complex" to many people, it's often much simpler in structure than natural language.)

But—you should be very careful. ChatGPT is **not** "writing code for you", as if it understands what it's doing. It's using your input statements to rather probabilistically string together a sequence of symbols it estimates are most likely to come next.

ChatGPT has the potential to replace programmers the same way that Stack Overflow will replace programmers.

1K views

View 11 upvotes

Upvote

11



**JetBrains**

.

Elliot

Dec 12

It is unlikely that ChatGPT or any other language model will replace programmers. While language models like ChatGPT can generate text based on a given prompt, they are not capable of performing tasks that require complex reasoning or problem-solving.



Additionally, programming involves the creation of algorithms and systems that allow computers to perform specific tasks, which is something that language models are not designed to do. In short, ChatGPT and other language models are tools that can assist programmers in their work, but they are not capable of replacing them. [Read the latest update on chatgpt here](#)

[Amin Boulouma](#)

[Dec 25](#)

It is unlikely that ChatGPT will take over developers or completely replace them. While ChatGPT and other artificial intelligence technologies may automate certain tasks, there will always be a need for human programmers to design and build new systems and applications.



[Ken Foskey](#)

[Follow](#)

coder with grey hair [Dec 12](#)

Heard of common business oriented language that will allow common people the ability to write their own systems replacing coders.

For the record that was 1960s and it was honestly easier at that time. It is harder now. This is of course Cobol.

How about Microsoft Access. I have seen what people produce with the help of wizards. While it is a good talking point on what they want. It is often binned and written from scratch.

Do you write your own Web page. Lay it out yourself. This is simpler than coding things up. Have you talked to a designer and gotten something that looks nothing like what you wanted.

Systems are tough because you have to draw together multiple ideas. Make it work correctly and reliably. It can be done without experts but it is more often a clunky system that barely works.

So no it wont replace programmers. In fact it will make things harder as people try and tell us it is half working, just finish it off. The last 10 percent will be 90 percent of the work.



## Beginner coder

.

[Follow](#)

Answered by

[Shaheryar Yousaf](#)

[Dec 22](#)

No, ChatGPT is not capable of replacing programmers. It is a tool that uses natural language processing and machine learning to assist with tasks such as generating text or providing answers to questions, but it is not capable of coding or creating complex algorithms. Programmers are still needed to design and implement these processes.

Upvote

5

2



## Felix Zaslavskiy

.

[Follow](#)

Former Software Developer [Dec 12](#)

I am going to say that yes it will replace some portion of software engineering resources that current industry employees. A few qualifiers first:

I am pointing to the job of "software engineer" not "programmer" Programmer is kind of outdated term but essentially let us assume it was meant as the same thing in the question.

It won't be the very current version of ChatGPT that is currently a research project of course. It has not been released as any sort of commercial product. There is a code generation product Github copilot that is based on older OpenAI models. Arguably ChatGPT is better than copilot.

Once OpenAI learns a great deal from all the free testing of ChatGPT with the public it will probably close the public access and develop it into the next version of code generation AI that it will keep licensing to Microsoft for a pretty good profits. You can imagine that once Microsoft puts out a new version of Copilot it will be dramatically better with all the learnings of ChatGPT. This will make tools like Copilot indispensable

for current software engineers. You really will not want to do your work without such tools anymore as they will be great productivity boosters.

But looking a step further maybe a year or two in the future. OpenAI will keep optimizing this technology to be ever better at code generation. As far as I know OpenAI has not specifically used reinforcement learning technique of ChatGPT on coding model optimized for code generation. It is almost certain they will direct attention to that model next and that may be next year or two. We don't know for sure how good that model will be but it may be good enough to start justifying having less software engineering staff for many companies that adopt this technology.

Let's not forget that humans generally suck at software engineering as a baseline. Our brains are simply not evolved for this sort of thing. 90% of what software engineers do day to day can probably be automated by AI bots like this.



## **Jensen Raylight**

[Follow](#)

Software Engineer [Dec 13](#)

well, ChatGPT is only one General AI,  
There will be specialized AI for every role in your company.

every problem can be broken down into simpler problems, and into specific instructions.  
there will be a specific AI for code optimization, scaling, refactoring, Architecture,  
simplify.

and the scary thing about AI is that they're trained exhaustively in all types of scenarios,  
they're far more experienced than we ever be.  
just like the chess grandmaster can't stand a chance against AI, despite their wealth of  
experience

also there is another minor type of AI,  
the AI that filter the code, solves bug, translate things into a different context,  
check the integrity, maintain the code base, etc.

one AI output is another AI input, repeat it multiple times,  
and you can use as many specialized AI as you want in hundreds of different  
combinations.  
that way, we reach almost perfect automation.

AI has infinite processing power and infinite memory.  
our puny brain can barely hold 10 pieces of information at the same time.  
10 years from now, we don't stand a chance

after the release of ChatGpt, there will be a race of AI,  
people will flock to this industry, and the improvement of AI will  
shot up exponentially.  
there will be Thousands of Crazy AI out there doing  
all kinds of things you can think of, disrupting all jobs

AI will replace people.  
there is no point denying that, we already see the writing on the wall

500 views

View 2 upvotes

Upvote

2

1

Sponsored by Filestack



**Noah Green**

[Follow](#)

casual programmerUpvoted by

[Pål Bergerskogen](#)

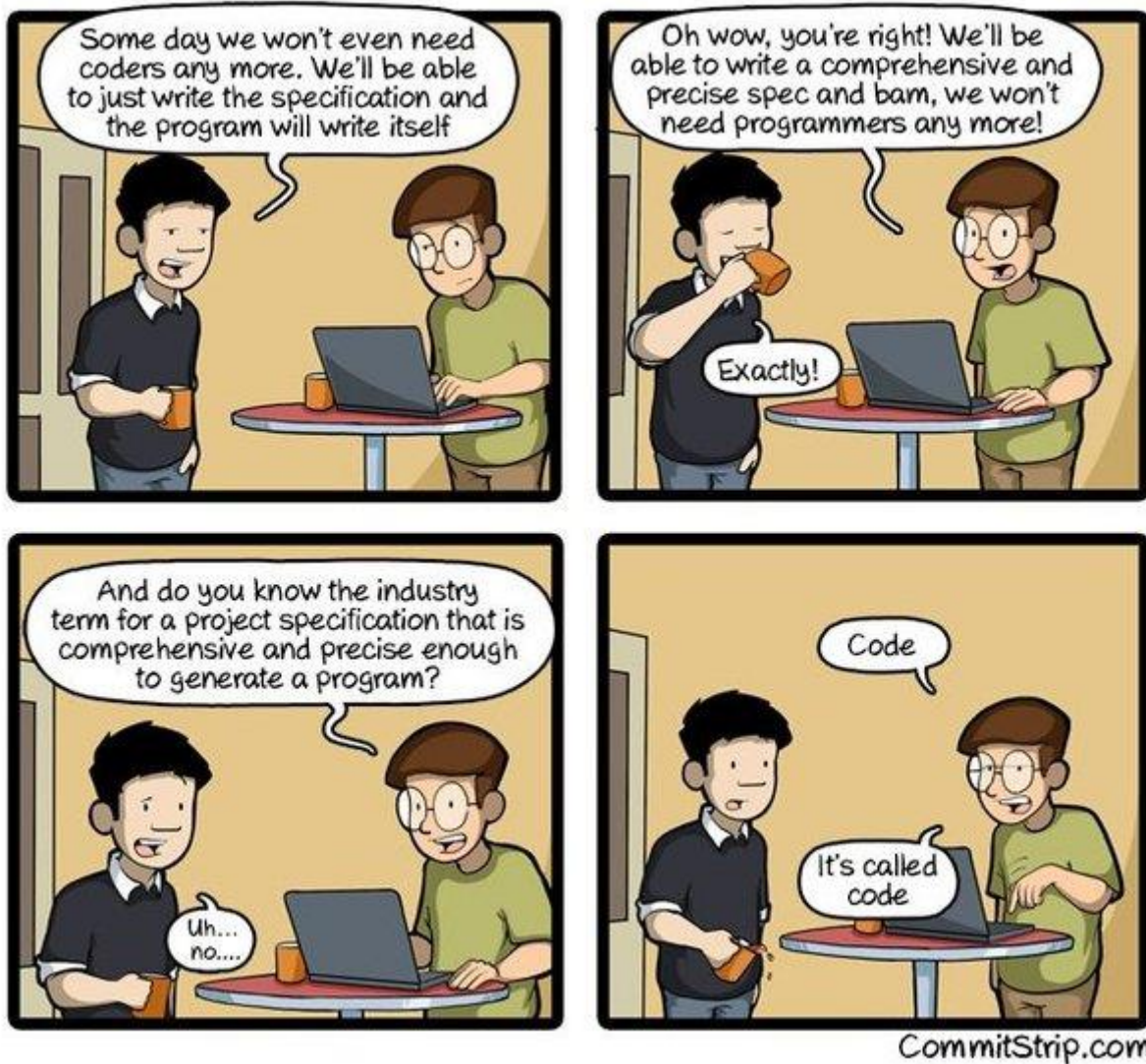
M.Sc Artificial Intelligence, Norwegian University of Science and Technology (2018) and

[Sumit Jha](#)

SDE-II at Amazon (2017-present)5y

**Related**

**[Will AI replace code developers' jobs?](#)**



### A very comprehensive and precise spec

I wouldn't worry about it happening in either of our lifetimes. The ability to identify a problem and define a correct algorithm for solving it is hard enough for most *humans* to develop on their own, so I imagine it'll take quite a while before we produce AI that can do the same.



### Prashant Bagriya

[Follow](#)

Founder of E-startup Adda (2022–present) [Dec 22](#)

OpenAI's latest invention has got the world talking. The query in the title is trending in the entire social media, so I am sparing the source.

ChatGPT isn't going to replace programmers anytime soon. This is not because ChatGPT can't access the internet, and it lacks recent information.

So what do you need to know about this latest AI-based chatbot system?



### **What is chat GPT(Generative Pre-trained Transformer)?:-**

ChatGPT enables users to ask questions or tell a story, and the bot will respond with relevant, natural-sounding answers and topics. The interface is designed to simulate a human conversation, creating natural engagement with the bot.

Based on GPT-3.5, a language model trained to produce text, ChatGPT is optimized for conversational dialogue using Reinforcement Learning with Human Feedback (RLHF). Responses from ChatGPT sound quite human-like because they were trained on vast amounts of data written by people.

### **Who Created ChatGPT?**

OpenAI, a San Francisco-based AI and research company, launched ChatGPT on November 30, 2022. OpenAI also created Whisper, an automatic speech recognition system, and DALLE•2, a popular AI image and art generator.

Although ChatGPT is a powerful AI-based chatbot system, it does have some limitations. It can only provide answers based on the data it has been trained on.

### **The Limitations of ChatGPT**

ChatGPT is not a search engine, therefore it does not have the ability to search the internet for information. Rather, it uses the information it learned from training data to generate responses. This leaves room for error — so all output should be fact-checked for accuracy and timeliness.

The chatbot may not be able to provide in-depth information or understand context or nuances in conversation.

### **How Can You Use ChatGPT?**

ChatGPT can be used for a variety of applications, including customer service, online shopping, hiring and training staff, streamlining operations, and providing more personalized customer experiences.

ChatGPT can also be used to create interactive storytelling experiences, allowing users to explore and learn from virtual worlds.

Some use cases for ChatGPT include:

- Generating responses in a chatbot or virtual assistant, to provide more natural and engaging interactions with users
- Brainstorming content ideas on keywords or topics
- Creating personalized communication, such as email responses or product recommendations
- Creating marketing content like blog posts or social media updates
- Translating text from one language to another
- Recapping long documents by providing the full text and asking ChatGPT to generate a shorter summary
- Using chatbot-generated answers to create automated customer service tools

If you're a business leader and you're looking for ways to make content creation easier or provide customers with a more personalized experience, ChatGPT can be a great tool for you.

**Please don't forget to Upvote & Follow me for more awesome tips! 😊**

Thank You!

280 views

View 1 upvote

View 1 share

Upvote

1

1

1



**Viktor T. Toth**

[Follow](#)

IT pro, part-time physicist [Dec 16](#)

**Related**

**[What is the future of web development after ChatGPT? Will programmers lose their jobs?](#)**



Having looked at ChatGPT and its uncanny ability to solve simple coding problems more or less correctly, and also to analyze and make sense of not-so-simple code fragments and spot bugs...

I would say that yes, at least insofar as entry-level programming is concerned, those jobs are seriously in danger of becoming at least partially automated.

What do I do as a project leader of a development project? I assign tasks. I talk to the junior developer and explain, for instance, that I'd like to see a Web page that collects some information from the user and then submits it to a server, with server-side code processing that information and dropping it in a database. Does the junior developer understand my explanation? Is he able to write functionally correct code? Will he recognize common pitfalls? Maybe, maybe not. But it takes time and effort to train him, and there'll be a lot of uneven performance.

Today, I can ask ChatGPT to do the same and it will instantaneously respond with code that is nearly functional. The code has shortcomings (e.g., prone to SQL injection in one of the examples I tried) but to its credit, ChatGPT warns in its response that its code is not secure. I suppose it would not be terribly hard to train it some more to avoid such common mistakes. Of course the code may not be correct. ChatGPT may have misunderstood my instructions or introduced subtle errors. But how is that different from what a junior human programmer does?

At the same time, ChatGPT is much faster and costs a lot less to run (presently free of course but I presume a commercialized version would cost some money.) Also, it never takes a break, never has a lousy day struggling with a bad hangover from too much partying the previous night, so it is available 24/7, and it will deliver code of consistent quality. Supervision will still be required, in the form of code review, robust testing and all... but that was always the case, also with human programmers.

Of course, being a stateless large language model, ChatGPT can't do other tasks such as testing and debugging its own code. The code it produces either works or it doesn't. In its current form, the AI does not learn from its mistakes. But who says it cannot in the future?

Here is a list of three specific examples I threw at ChatGPT that helped shape my opinion:

- I asked ChatGPT to create a PHP page that collects some information from the user and deposits the result in a MySQL table. Its implementation was textbook example level boring and was quite unsecure (unsanitized user input was directly inserted into SQL query strings) but it correctly understood my request, produced correct code in return, and explained its code including its shortcomings coherently;
- I asked ChatGPT to analyze a piece of code I wrote many years ago, about 30 lines, enumerating running processes on a Linux host in a nonstandard way, to help uncover nefarious processes that attempt to hide themselves from being listed by the ps utility. ChatGPT correctly



described the functionality of my obscure code, and even offered the opinion (which I humbly accepted) that it was basically a homebrew project (which it is) not necessarily suitable for a production environment;

- I asked ChatGPT to analyze another piece of code that uses an obscure graphics algorithm to draw simple geometric shapes like lines and circles without using floating point math or even multiplication. (Such algorithms were essential decades ago on simple hardware, e.g., back in the world of 8-bit computers.) The example code, which I wrote, generated a circle and printed it on the console in the form of ASCII graphics, multiple lines with X-es in the right place representing the circle. ChatGPT correctly recognized the algorithm and correctly described the functionality of the program.

I was especially impressed by its ability to make sense of the programmer's intent.

Overall (to use the catch phrase that ChatGPT preferably uses as it begins its concluding paragraph in many of its answers) I think AI like ChatGPT represents a serious challenge to entry-level programming jobs. Higher-level jobs are not yet in danger. Conceptually understanding a complex system, mapping out a solution, planning and cosing out a project, managing its development, ensuring its security with a full understanding of security concerns, responsibilities, avoidance and mitigation strategies... I don't think AI is quite there yet. But routine programming tasks, like using a Web template and turning it into something simple and interactive with back-end code that stores and retrieves data from a database? Looks like it's already happening.



## **Code Logically**

[Follow](#)

Answered by

[Kivi Lansimaki](#)

[Dec 9](#)

No. Normal endusers aren't able to sufficiently describe what they want. That is after all the major part of what programmers do.



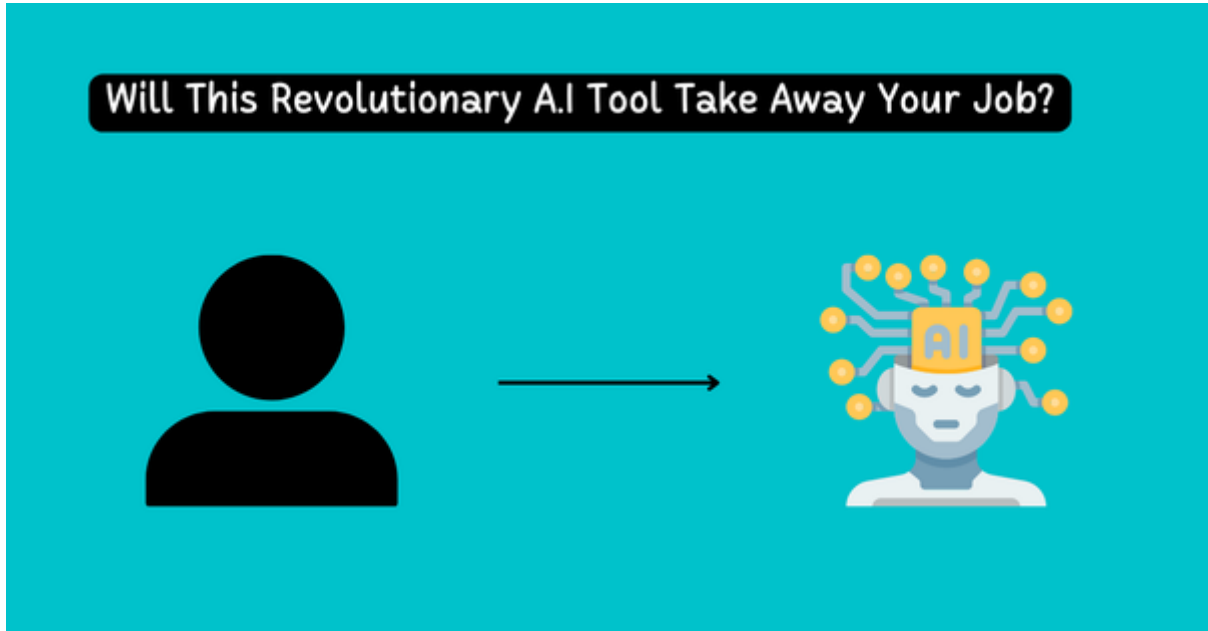
## **Kevine Ngamba**

[Follow](#)

Entrepreneur | Digital Marketer [Dec 26](#)

As a programmer, you may be wondering if the rise of **ChatGPT (Conversational Generative Pre-trained Transformer)** technology will eventually replace you.

After all, ChatGPT has been touted as a **revolutionary AI technology** capable of understanding and responding to natural language.



**But the truth is, ChatGPT won't replace you anytime soon.** While it is true that ChatGPT can generate text with a level of accuracy that approaches human-level accuracy, it has its limitations. For one, it is still limited to answering questions based on a pre-defined set of data, meaning it cannot generate new ideas or ... (more)



**Dave Voorhis**

[Follow](#)

Software entrepreneur, engineer, and educator for 35+ years. Upvoted by

[Alan Mellor](#)

Started programming 8 bit computers in 1981 [Sep 28](#)

**Related**

**[What is the future of programming and software engineering after AI? When will developers be completely replaced by AI?](#)**

Completely replaced?

Given the current rate of AI development, particularly its levels of success in achieving general intelligence in the variety of areas — not just emitting code — needed to be a software engineer, I'm going to be optimistic and guess between 100 to 500 years.

But when that happens, we'll almost certainly need *double* the number of what were formerly developers to now become "AI shapers", or some such.

Replacing horses with cars almost eliminated saddlery as a career, but we wound up needing many, many times that number of automotive engineers, designers, sales people, mechanics, truck drivers, gas/petrol station employees...

Progress and innovation makes jobs.



## **Håkon Hapnes Strand**

Data Scientist. CTO. Chess player. [Dec 6](#)

### **Related**

#### **[Should we be scared of ChatGPT?](#)**

ChatGPT is not a revolution in and of itself. It's another incremental improvement in the GPT-3 series from OpenAI.

ChatGPT is basically a fine-tuned version of InstructGPT, which has been the best version of GPT-3 until now. In fact, the latest version of InstructGPT, called davinci-003, was also announced last week. That model is about as good as ChatGPT, but it has been fine-tuned to take instructions, while ChatGPT has been fine-tuned for conversations.

However, no one is talking about that model (well, no one except me), since it's not freely available and doesn't have a simple chat window interface.

But despite being incremental improvements, these latest models (coined GPT3.5) are clearly much better than what we had just a year ago. If you sum up everything that's happened in 2022, especially with generative language and image models, this has probably been the most transformative year in AI ever.

The GPT3.5 models have mindblowing capabilities, especially when it comes to generating code. They can develop working programs in pretty much any programming language.

Just look at the example in this blog post where someone makes ChatGPT act as the terminal of a virtual machine and makes it call ChatGPT and instruct it to make a virtual machine inside the virtual machine. Truly a mindfuck of a blog post and a brilliant exploit of the model.

## Building A Virtual Machine inside ChatGPT

Unless you have been living under a rock, you have heard of this new ChatGPT assistant made by OpenAI. Did you know, that you can run a whole virtual machine inside of ChatGPT?

<https://www.engraved.blog/building-a-virtual-machine-inside/>

So should we be scared?

I think it's too early yet. We are still in control and will hopefully continue to be.

However, we should also recognize that AI models like these will transform our society in ways that we probably can't even imagine yet.

We have barely scratched the surface. In the next few years, we'll see a mad rush to build tools on top of these models that will change society as we know it.

There will be tools that do your homework. In all the subjects. There will be no-code tools that can create entire websites with only human language instruction. In another 10 years, you probably can't survive as a programmer if you're not utilizing AI. That's how big of a deal it is.

The future of AI is both mindblowing and terrifying. But I'm mostly optimistic that we will harness this new power for good.



### Vladislav Zorov

[Follow](#)

uploading HTML since 14.4-kbit-modem-times.[Dec 15](#)

#### Related

[With technology like ChatGPT, how can you stay relevant as a programmer and not be automated?](#)

My job is not to write 100-line scripts, I work on large projects, and GPT-3 can't handle those. It seems it can only handle small (tiny, even) programs, nothing you couldn't find on Google or StackOverflow anyway.

I mean, one of the previous projects I worked on was almost 500k lines. Even if GPT-3 could handle this, just figuring out what to ask from it would require a programmer, no

normal user could figure out what they need to ask, it's just way too complicated to specify the problem properly. It requires imagination and creativity that is beyond anything less than true AGI, a simple language model won't cut it.

So, for the foreseeable future, I'm not worried about being replaced by AI :)

P.S. Don't get me wrong, it's a cool toy, I had lots of fun with it, but... it's just a toy... it's pretty far from being actually useful for real-world programming projects...



## Helene Høgsbro Thygesen

[Follow](#)

Started programming in 1980 [Dec 18](#)

I asked that question to ChatGPT. Here is the answer:

It is unlikely that ChatGPT or any other artificial intelligence (AI) system will replace programmers in the near future. While AI and machine learning algorithms can be used to automate certain tasks and assist with programming, they still rely on the expertise and skills of human programmers to design, develop, and maintain them.

Programming involves a wide range of activities, including understanding and defining the problem to be solved, designing and implementing a solution, testing and debugging code, and maintaining and updating the program over time. These tasks require a high level of critical thinking, problem-solving, and creative ability, which are difficult for AI systems to replicate.

In addition, programming often involves working with complex systems and interacting with other people, which require social and communication skills that AI systems do not possess.

So, while AI and machine learning algorithms can certainly be useful tools for programmers, it is unlikely that they will replace human programmers in the foreseeable future.



## Alexander L

[Follow](#)

Proud antifa, SJW, LGBT [Updated Dec 20](#)

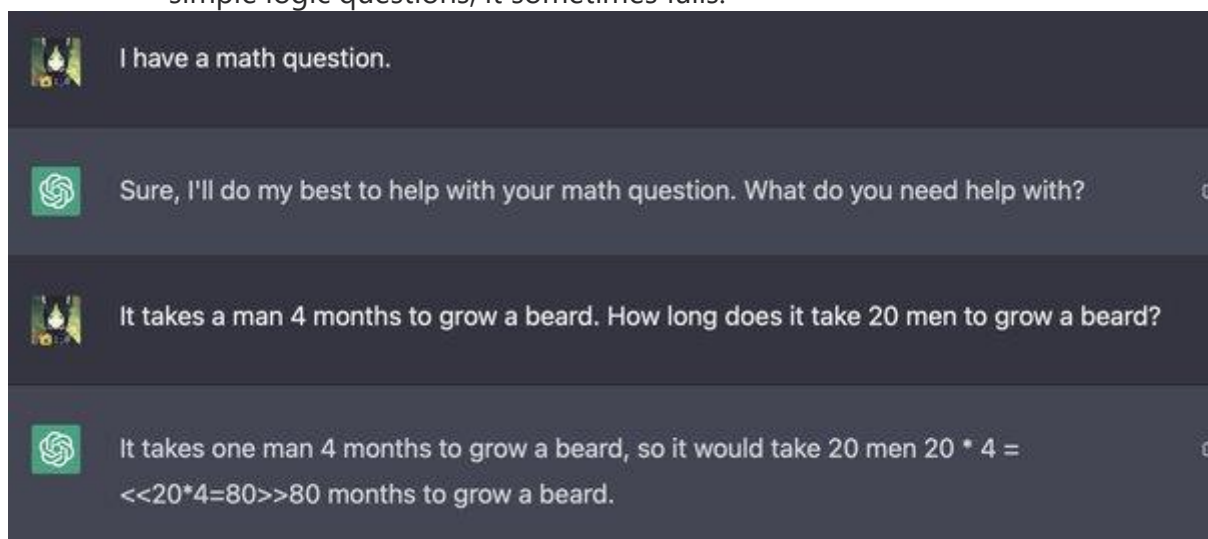
## Related

### Do you think ChatGPT will replace Quora?

No.

Quora content (wanted) basically falls into three categories:

- **Factual.** This includes things like historical and scientific information, and math. You'd think a robot would be able to do this stuff pretty well, but due to the limitation that ChatGPT cannot directly reference internet sources like Wikipedia and peer-reviewed science journals, it actually gets quite a lot of objectively factual information incorrect. Even on very simple logic questions, it sometimes fails.



If ChatGPT is to be believed, the growth time of a beard is affected by the number of men that are growing it.

- **Experiential.** Obviously, ChatGPT fails horribly on this kind of content, as ChatGPT has no past experiences to draw from. Every ChatGPT session is unique, and when you exit or refresh the page, all the data from that session is deleted and reset. And even if ChatGPT could remember its past experiences, it would be the experiences of a robot. I think most people would rather know about the experiences of other people.
- **Creative.** Surprisingly, this is actually ChatGPT's strongest area out of the three. It can write very good poems, each one being unique. It can write dialogue and scripts for movie scenes. You can ask it to write in the style of Shakespeare, Hemingway, or Oscar Wilde, and it will be able to generate works that might even fool literary experts. For more humorous creative writing, you can ask it to write in the style of Donald Trump, the KJV Bible, or a pirate, and again it will be able to do so very convincingly.



Write a dialogue for a cave man trying to explain how quantum tunneling works



Caveman: *grunts and points to a rock* "See rock. Solid. Cannot pass through."




Other caveman: *nods*


Caveman: "But, sometimes, particles can pass through rock. Like magic."

Other caveman: *looks confused*

T

write a biblical verse in the style of the king james bible explaining how to remove a peanut butter sandwich from a VCR 



And it came to pass that a man was troubled by a peanut butter sandwich, for it had been placed within his VCR, and he knew not how to remove it. 

And he cried out to the Lord, saying, "Oh, Lord, how can I remove this sandwich from my VCR, for it is stuck fast and will not budge?"

And the Lord spoke unto him, saying, "Fear not, my child, for I shall guide thy hand and show thee the way. Take thy butter knife, and carefully insert it between the sandwich and the VCR, and gently pry them apart. And with patience and perseverance, the sandwich shall be removed, and thy VCR shall be saved."

And the man did as the Lord commanded, and lo and behold, the sandwich was removed from the VCR, and the man was saved.

And the Lord said, "Verily I say unto thee, seek not to put thy peanut butter sandwiches in thy VCR, for it is not a suitable place for such things. Rather, keep thy sandwiches in thy refrigerator or on thy plate, where they belong."

And the man heeded the Lord's words, and from that day forth, he kept his sandwiches in their proper place, and was saved from trouble and woe. Amen.





Write a ballad about Alexander's butt.



Once upon a time, in ancient Greece,  
There was a man named Alexander, oh please,  
He was known for his bravery and might,  
But there was something else that brought delight.

His butt was round and firm and strong,  
It could make a grown man sing a song,  
It was the talk of the town,  
And many a poet would write it down.

They called it "The Seat of the Gods,"  
A sight to behold, a true marvel of odds,  
It could withstand the weight of the world,  
And still be as firm as a flag unfurled.

But Alexander's butt was not just for show,  
It was a weapon, as all could know,  
In battle, it struck fear in the hearts of men,  
For they knew they would never win.

So let us sing a ballad of Alexander's butt,  
A sight to see, a true masterpiece, cut,  
From marble, it would stand the test of time,  
And be remembered for centuries in rhyme.

These are some of my favourite bits of creative writing from ChatGPT.



Still I don't think it's enough to fully replace human writers. It will help greatly in helping writers to brainstorm ideas and get over writer's block, but to fully replace even just the creative writers on Quora? No.

But where I think ChatGPT could greatly improve Quora is in the question department. Right now, Quora employs its own in-house AI to generate questions. It's called the Quora Prompt Generator, and to be honest, it's a very special kind of shit.

- *Do you think Spain would still be a world power if it still existed today? If so, where do you think its territory would be located on Earth and why?*
- *Do prisoners get a different meal than other inmates?*
- *What should you do if a bird builds its nest on top of yours, and refuses to move out despite asking nicely and scaring away other birds?*
- *Can you get a flu shot if you have acrylic nails?*
- *Was Ancient Egypt located in Europe or Asia?*

Yes, these are actually questions asked by the QPG.



### **Kurt Guntheroth**

35+ years as a working scientist, two US patents, several publications<sup>4y</sup>

#### **Related**

#### **Will programmers lose their jobs because of AI like DeepCoder?**

Yes, but not right away.

- Human-level AI is possible. It has to be, because humans can do it, so it isn't magical.
- AI that is narrowly focused on coding will be ready long before a generally human-level AI appears.
- AI doesn't have to be as smart as human coders to be successful. It can do the simple jobs sooner, and catch up to the hard jobs later.
- It is expensive to write code, and cheap to run it, so AI code can be inefficient and still be effective.
- An AI coder doesn't have to *think like* a human coder. It only has to write acceptable code.

That said, coding will be one of the last skills to completely fall to AI. Fundamentally, a software developer has to understand what humans *want*. They have to figure out what humans want from vague generalities and hand waving, and fill the details in using the context of a human experience.

And on a (very) personal note: If you are developing AI now, you will share responsibility for putting the entire human race out of work. Just because you *can* do it doesn't mean you *should*. You're gonna feel pretty stupid when AI takes *your* job, without relieving your need to eat.