

# Am I A Good Programmer?

Kate Gregory

kate@gregcons.com  
@gregcons on Twitter

Programmer

Good

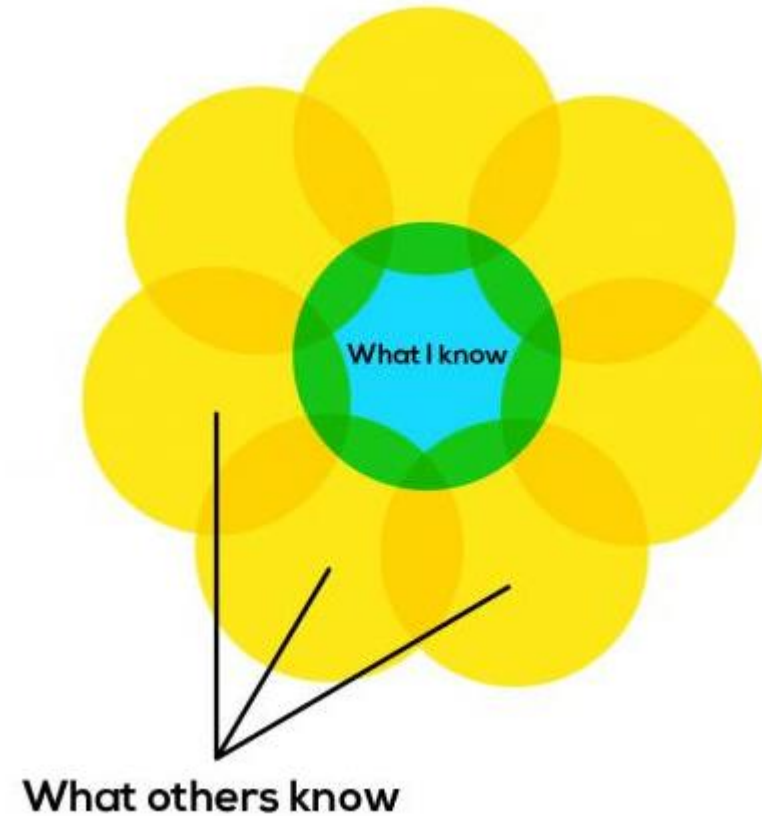
I

Am

# Dunning–Kruger effect

- Not all skills!
- Your lack of skill needs to cause a lack of discernment as well
  - Sense of humour
  - Perfect pitch
- I can easily assess if I can run fast or throw a ball far

# Imposter Syndrome



# Learning and Imposter Syndrome

- Learning and growing is good
- Never doing anything new? You're not learning and growing
- Regularly trying something you don't know yet?
  - You're not an imposter, you're a learner!
  - "I can't do this!"
    - "...yet"
- While some learning includes discomfort, not all does
  - And being uncomfortable doesn't prove you're learning



# Is Wondering Actually Enough?

- Are the good programmers just the programmers who care about being good?
- Is wanting to be a good programmer a kind of proof you are one?

# Bad Programmers Exist

- This isn't "No Bad Dogs" –land
- Some people will do better in a different job
- Some of those people really want to be good at this
  - But they're not
  - (yet?)
- It helps to want to improve, but improvement is still a necessity



**Programmer Humor**

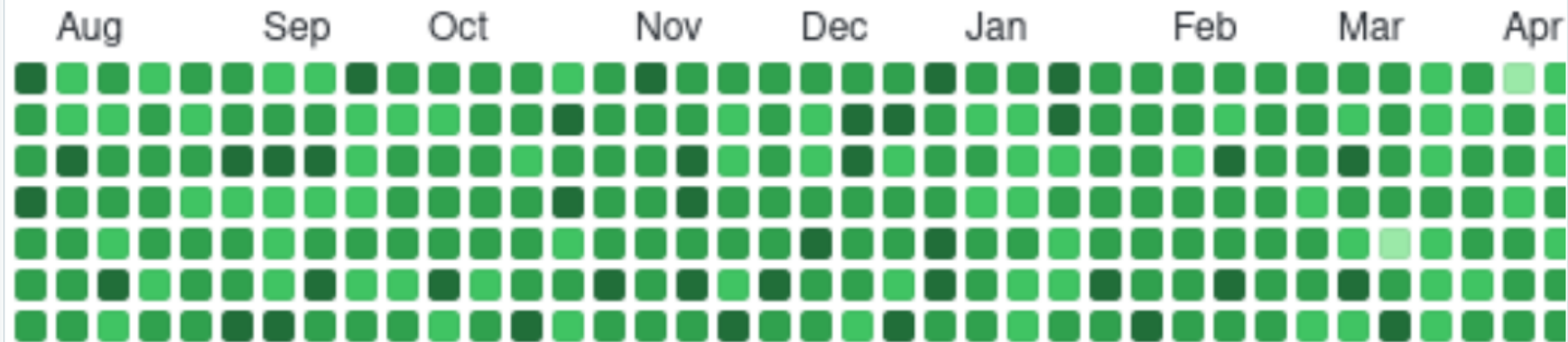
@PROGRAMMERHUMOR



What's stopping you from coding like this!?

[reddit.com/r/programmerhu...](https://reddit.com/r/programmerhu...)

ns in the last year



count contributions

Less

**NEW!** [View your contributions i](#)

9:00 AM · Jun 17, 2022 · programmerhumor

# Non metrics

- Github records
- Who has employed you
- Time as an independent
- Age and experience
- Certifications, diplomas, ...

# How Can You Know?

- Have you avoided (or minimized) the most outrageous catastrophes?
- Do other people ask your advice?
- Do you enjoy it?

# Outrageous Catastrophes

- Not crashing production, not deleting a database
- Realizing after months of work that some fundamental assumption was completely wrong and too much depends on it
- Creating something unmaintainable that nobody dares to touch
  - Double edged sword
- Failing to consider or test something vital (like performance or scaling) and only discovering issues after delivery
- Leaving half done projects in your wake

# Others ask your advice

- Directly, one on one
- In places they know you will see it
  - Slack, discord, group chats
  - StackOverflow
  - Mailing lists
- Do they take your advice when you give it?
  - Do they come back for more?
- People follow you on Twitter or read your blog
  - Do they recommend you to others?

# You enjoy it

- First, ask yourself
  - Is this fun?
  - Would I do it if I didn't need the money?
- Second, ask your code
  - Does it look like it was written by someone happy?
- Third, ask your projects
  - Do they get finished?
  - Are they complete? (Tests, documentation, ...)
- Fourth, ask your resume
  - Does it glow with joy?





# If you're not sure you're good

- Get better!

# How Can You Get Better?

- Attend conferences
  - At least watch some talks online
- Read blogs
- Listen to podcasts
- Read Twitter
- Read code
- Read books
- Take courses

# Conferences

- Accu.org
  - Cpponseas.uk
  - Cppnorth.ca ← we are here!
  - NDCtechtown.com
  - Meetingcpp.com
- 
- <https://isocpp.org/wiki/faq/conferences-worldwide>

# Blogs

- [www.fluentcpp.com](http://www.fluentcpp.com) – Jonathan Boccara
- [www.modernescpp.com](http://www.modernescpp.com) – Rainer Grimm
- [www.foonathan.net](http://www.foonathan.net) – Jonathan Müller
- [thephd.dev](http://thephd.dev) -- JeanHeyd Meneide

# Podcasts

- [Cppcast.com](http://Cppcast.com) – over now, but catch up old episodes
- [www.youtube.com/user/lefticus1](http://www.youtube.com/user/lefticus1) – Jason Turner C++ Weekly
- [adspthepodcast.com](http://adspthepodcast.com) – Bryce Lebach & Conor Hoekstra
- [www.twoscomplement.org/#podcast](http://www.twoscomplement.org/#podcast) – Matt Godbolt and Ben Rady

# Twitter

- [twitter.com/shafikyaghmour/lists](https://twitter.com/shafikyaghmour/lists) – the C++ one is a great start
- [twitter.com/isocpp](https://twitter.com/isocpp) -- feed of posts from others
- When someone you follow retweets something good, follow that new person
- Don't try to read it all
  - Step in the river for a while
  - The river will be there again later
- Great way to find blogs, podcasts, conferences, ...

# Consuming Large Content

- Code
  - Reading good code shows you what good code does
  - If you're confident, reading bad code can also be very enlightening
- Books
  - C++ ones abound (get recommendations though, many are bad)
- Courses
  - Cost real money
  - I think they're worth it



# Using

- Someone explains – now you understand
- You discover a technique or pattern – start using it
- You discover a library – start using it
  - Save time
  - Less bugs
  - “No raw loops”
- You find a great place to ask questions - ask there next time
- You discover a source of great information – monitor it

# Find a Community

- Support and encouragement
- Practical help
- News and announcements
- A place for you to practice helping



**Scott Hanselman** 🇺🇦 ✓

@shanselman



**abdoulaye diallo** @abdoula55769092 · Jun 22

@shanselman this is a long shot but. How do I get better as a programmer? I'm 1 years into my career but i don't feel like I'm learning as fast as i could. Most tickets i get i need some assistance getting unblocked. My stack is event driven architecture

2:05 AM · Jun 22, 2022 · Twitter for iPhone



**Scott Hanselman**  

@shanselman



First, welcome to tech and congrats on your year!



**abdoulaye diallo** @abdoula55769092 · Jun 22

@shanselman this is a long shot but. How do I get better as a programmer? I'm 1 years into my career but i don't feel like I'm learning as fast as i could. Most tickets i get i need some assistance getting unblocked. My stack is event driven architecture

2:05 AM · Jun 22, 2022 · Twitter for iPhone



**Scott Hanselman** 🇺🇸

@shanselman



First, welcome to tech and congrats on your year!

Second, keep notes of things to study. Surprisingly few folks keep track of what to learn.



**abdoulaye diallo** @abdoula55769092 · Jun 22

@shanselman this is a long shot but. How do I get better as a programmer? I'm 1 years into my career but i don't feel like I'm learning as fast as i could. Most tickets i get i need some assistance getting unblocked. My stack is event driven architecture



**Scott Hanselman** 🇺🇸

@shanselman



First, welcome to tech and congrats on your year!

Second, keep notes of things to study. Surprisingly few folks keep track of what to learn.

Third, study groups, lean on friends. You can't do this by staying yo later.



**abdoulaye diallo** @abdoula55769092 · Jun 22

@shanselman this is a long shot but. How do I get better as a programmer? I'm 1 years into my career but i don't feel like I'm learning as fast as i could. Most tickets i get i need some assistance getting unblocked. My stack is event driven architecture



**Scott Hanselman** 🇺🇸 ✓

@shanselman



First, welcome to tech and congrats on your year!

Second, keep notes of things to study. Surprisingly few folks keep track of what to learn.

Third, study groups, lean on friends. You can't do this by staying yo later.

Fourth, be patient. You'll be in tech for 30 years!



**abdoulaye diallo** @abdoula55769092 · Jun 22

@shanselman this is a long shot but. How do I get better as a programmer? I'm 1 years into my career but i don't feel like I'm learning as fast as i could. Most tickets i get i need some assistance getting unblocked. My stack is event driven architecture

# How Can You Get Better?

- Speak at conferences, user groups etc
- Write a blog
- Tweet
- Write code
  - And discover the consequences of what you write
  - Pairing and mobbing to learn how others think, or receive what you think
- Appear on a podcast
- Teach more formally
- Write a book



# Caring about code

- Abstraction
- Performance
- Naming
- Maintainability and expressivity
- Reviewing other people's code
  - To make it better
  - To make yourself better
- Library and API design
- Architecting related systems

# Caring about More than Code

- Estimation, progress measurement, prediction
  - To support managing the effort and related tasks eg deploys
- Testing
- Documentation
- Build systems
- Source control
- Third-party libraries
- Standardization

# Caring about other programmers

- Teaching (formally)
- Broadcast sharing (blogs, tweets, videos, talks...)
- Mentoring
- Code Review
- Writing code that helps them
  - Great examples
  - Reusable
  - Readable
- Become the person other people turn to

# How Can You Know?

- Have you avoided (or minimized) the most outrageous catastrophes?
- Do other people take your advice?
- Do you enjoy it?