

[Domains](#)[Contests](#)[Leaderboard](#)[Submissions](#)

beginner_007 ▾

75
Shares

18

[← All posts](#)

Exclusive Q&A with World-Renowned Coder SnapDragon

coding



Derek Kisman is among the most accomplished and experienced competitive coders around. Growing up in Canada, Calgary, he's a self-taught competitive coder who started coding when most of us play with legos!

In the 80s, while schools were still trying to figure out how to teach computer science in schools, Kisman was drawn to coding and competitions purely out of personal determination and passion. Because of a lack of accessibility in Canada to international contests during his high school years, Kisman's very first programming contest was the tryouts the ACM team at his alma mater, University of Waterloo. The limited access to competitive coding resourced didn't stop Kisman from constantly pushing himself. In fact, he actually had to write his own challenges because there weren't enough coding challenges out there!

Through intense dedication and thousands of hours of practice, Kisman consistently ranked as a top coder for several years, among many other accolades. Little did he know he would now be serving as a judge at the prestigious ACM International Collegiate Programming Contest (ICPC)!

Today, Kisman is still a strong coder and competitor. Most recently, he won second place at the CodeSprint ZenHacks (or hackathon) sponsored by the fast-growing startup Zenefits. Today, Kisman works as a developer at Zenefits!

We are very excited to announce our second live streaming 6-hour Live Q&A event in which Kisman discusses tackling the toughest challenges featured at ACM-ICPC 2015 in addition to some HackerRank challenges. Just about 3,000 passionate coder are going to be tuning in! Are you? In the meantime, here's the exclusive interview with SnapDragon:

75
Shares

18



Source: *KupasKode*

When did you start coding and how did you learn?

I started coding around age 5 or 6 on a TI-99/4a. We upgraded it to 16 kilobytes of memory so it could run Logo. 😊 So I've been programming most of my life and I'm mostly self-taught. In university, of course, I spent a LOT of time practicing algorithm contests.

When was your first programming contest?

Canada was just entering the IOI when I graduated, so I didn't have the chance to try out for it. My first contest was a qualifier for the ACM team in my first year at Waterloo. I didn't really know what I was doing, but I made it onto a team ... just barely!

How did you become such a skilled competitive programmer?

Lots and lots of practice. Having a solid math background helped too, but mostly it was spending thousands of hours just solving ACM Regionals problems UVA problems (which was basically the only online judge portal back then). Our coach, Gordon Cormack, would spend a lot of time setting up the practices and discussing solutions with us afterwards.

What was the biggest hurdle on your road to where you are today?

Getting old, I guess? I haven't faced too many hurdles. I got pretty steadily better for a long time, until I peaked in the early TopCoder years. Since then I've spent a lot less time practicing and I've gotten a little slower.

What is the technology & programming scene like where you grew up?

Growing up in Calgary, there was a school programming event every year where kids would go to show off their projects. I entered every year and I almost always bit off more than I could chew with my projects. 😊 When I went to the University of Waterloo is when my contest training really started. Waterloo is very strong in the ACM scene, and there were a lot of motivated people there to learn and discuss problems with.

What do you think is the biggest mistake competitive programmers make?

75
Shares

“The biggest mistake programmers make is probably getting discouraged and not practicing enough. Or practicing only on problems they feel they're good at.”

- Derek Kisman, aka SnapDragon

18

Probably getting discouraged and not practicing enough. Or practicing only on problems they feel they're good at. I've always thought it was important to be able to tackle any problem that comes my way. So, I specifically practiced areas (like Geometry) that I was weak at. Even when I do come across a problem I can't solve, I try to look for what I'm missing and imagine how I would have solved it on a better day.

How often do you practice programming on any given day & weekend?

These days I don't practice much. It probably averages out to a couple hours a week, mostly helping out with test solving contests. So I'm not as good as I used to be! Back in my serious years, I would solve multiple 5-hour contests a week, plus many hundreds of problems on the side. Good thing I enjoy algorithm contests so much...

What language is by far the best, most useful in 2015?

“I still use C++ (which will never truly go out of style).”

- Derek Kisman, aka SnapDragon

Sadly, probably Javascript, though it's a mess of a language. I still use C++ (which will never truly go out of style), but I'm a bit of a dinosaur. Python and C# do seem to be the "cool" languages today. Fortunately, if you know one language it's usually pretty easy to pick up and move to another. You'll already have a good mental model of what's going on behind the code.

What language is by far the *least* useful?

I have a soft spot for languages specifically designed to be useless, like Brainfuck and Intercal. 😊 In terms of real languages... well, I've always hated Java. It's slow and makes a lot of common operations annoying. I can't really say it's "useless," though, since a lot of companies still use it.

What factors do you believe help you win?

Experience. So many of the algorithm problems out there are just plain similar to ones I've solved in the past. (It's really, really hard to come up with truly unique problems). Coding something for the 5th time is always going to be faster than coding it for the 1st. The other key skill is to learn to write code elegantly and cleanly. Ideally the code should map very simply to your mental model of the algorithm, making comments almost unnecessary and ensuring that any bugs

75
Shares

18

and special cases stand out. Since programming contests are typically extremely strict (running your code on secret data and giving you nothing but a "pass" or "fail"), it's important that your code be nearly bug-free when you first write it. If you can comprehend your entire program just by glancing over it, it's much more likely to be correct.

Why do you choose to compete on HackerRank?

“ I think HackerRank does the best job of "gamifying" programming. Solving a task, getting that little checkmark and progress towards a badge... it's pretty addictive! ”

- Derek Kisman, aka SnapDragon

I came for ZenHacks and stayed because I really like the interface. Out of all the sites I've competed on, I think HackerRank does the best job of "gamifying" programming. Solving a task, getting that little checkmark and progress towards a badge... it's pretty addictive!



127
KUDOS

More from my site



War, Passion & the Origin of Computer Societies



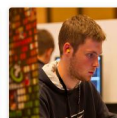
Why Don't More CS PhD Breakthroughs Turn into Companies?



The History of 'Hello, World'



The Real Reason Women Only Earn 18% of CS Degrees



Exclusive Q&A with Mimino: Inside the Mind of Pro Competitive Coder



How Mimino Solved 78 Project Euler Problems in Under 24 Hours

75
Shares

18

4 Comments

HackerRank

1 Login

Recommend

Share

Sort by Newest

Join the discussion...

Puneeth Shivalingaiah

7 days ago

Excellent article :) Thanks SnapDragon

1 ^ | v • Reply • Share

Dmitry Kamenetsky

9 days ago

Great article, but please run a spell check before publishing things.

^ | v • Reply • Share

tejesh papineni

12 days ago

Nice article. @Ritika Trikha Are you sure its additive?

1 ^ | v • Reply • Share

humblemanview

12 days ago

nice article inspiring :)

1 ^ | v • Reply • Share

Subscribe

Add Disqus to your site

Privacy

Company

About Us
Careers
Blog

Docs

Scoring
Environment
FAQ

Other

Privacy Policy
Freenode
Contact Us



© 2014 Interviewstreet, inc.

[Facebook](#) - [Twitter](#)