The following are a list of interesting quotes from the interviews. I simply selected the quotes that were most intriguing to me and organized them by topic.

Starting over

And I don’t wanna rewrite it, I just want to, like, find [the error]. - F2

I put the bug in, so I know I can take it out. At worst, I just, like, delete it and start over. - F2

Learning to debug is intuitive

Debugging techniques are obvious

It’s just kind of obvious. I don’t know. There’s no like method, like, to what’s wrong. I don’t know. It’s just all just thought process. - F1

People figuring out debugging on their own

I think it’s important that it’s taught in school more rather than just kind of, “go figure it out”. Because it’s such a critical part of programming, I feel like it’s not emphasized enough. Everyone just kind of figures it out. So, that’s just what I find that everybody struggles with, about debugging. It seems like everybody is reinventing the wheel in a lot of ways, especially when it comes to complicated scenarios. There’s not like a common knowledge. You know? Here’s how you debug, you know, these scenarios versus those scenarios versus performance versus memory versus, you know? Those kind of things versus UI. - P3

When talking about a new debugging tool: I just played around with it. I looked for what I needed until I found it. Now, there’s probably more there that I could find useful. - P2

We’re probably all the same. We had to probably kinda learn it on our own. Because it’s hard to generalize, because I’ve been trying to answer these [questions] in a general fashion, but yet, I’m thinking of specific examples. But, I know that overtime the way I go about it is the same. - P2

People have difficulty describing techniques

It’s second nature to me now. It’s more of… the hardest part is, kinda, I think maybe knowing where to start. I think is, at least in the students I have seen, they don’t know where to start looking. - P2

I doubt I know every technique, and I’m sure if I saw other people [debugging] or took a class I could probably improve. I’m sure. - P2

No one trained me specifically, and so, I suppose I don’t know how to train anyone else. And I don’t know that I have a good technique or not. I’m able to do it, but whether I have a good technique or not, I don’t know. But not having been taught specifically, I don’t know how that I would teach it because it has become a kind of part of me. And so is what I am doing… would it make sense to anyone else? You know? Or can I whittle the things that I do down to components that someone else would understand, or is it just my twisted mind? I don't know. I probably could, but it's not something... I haven't actively taught anyone. - P2

Debugging is not as intuitive as we might think

I noticed that certain numbers were sequential, so, I was like, "those are probably the rows," and I checked it out. But someone had to show me where the rows were labeled in the window because that's, just of, like, hidden at the bottom right, so I couldn't see that. Before I just had to, like, guess. - F6

Some techniques are difficult to learn

People struggle to understand the usefulness of techniques they don't use

I haven't really done the whole "sprinkling with couts" thing very much. I don't know why. For some reason that seems like more work to me. It probably isn't in the long run, but I really haven't done it very much, so I don't know. - F4

The whole error messages, um, still being greek is, a little bit... I do feel, kind of, out of my depth sometimes just because of that. - F4

I feel like with a lot of really powerful tools, there's kind of like a barrier you have to break through, and that is, the barrier is, knowing which questions to ask. I feel like especially young computer science students, don't know what question to ask to start off, so they don't even know how to get going. - J3

When asked why she doesn't use breakpoints: I'm sure it's easier, but it's more of that getting over the hump of learning it to then use it instead of just using what I know and going from there. - Se3

There have been times when I've thought, "It'd be really nice to use a debugger right now," but I don't take that opportunity to teach myself, or I think... I'm just kind of getting over that hump of learning... that learning curve would be too long for a certain problem. And so, I think ultimately, I just need to, like, hunker down and figure out a good debugger. Just so... I think it would save me a lot more time in the long run. - Se3

Sometimes it's harder to use, like, the little, like, breakpoints and all that stuff. - So3

People use techniques because they are easy and familiar

I use a lot of print statements. I think that's my primary go to, which is not always the best, I know, but it's kind of the easiest - Se2

For some reason I have the idea that… If I feel like it's just a simple issue, then I'll just do a simple print statement, I don't know, just because I'm more comfortable with doing it that way. I don’t think I really have a great reason other than I’m just more comfortable, or it’s what I’m used to. - Se2

Well, let me try and do the easy stuff and see if it fixes. I don’t really want to have to go through all of my code and work it out through my mind. I just want to try and find the easy fix. If I have to get to that point, I’m not very confident about it. I mean yeah I could probably work it through, but I choose not to a lot of the time. I try and find the easy way out. So am I confident in debugging? No, I... No, if I knew how to do it well I think it would be so much easier. - J2

Attitudes about debugging

Debugging shouldn’t exist

It should have worked. It has to work. I’m smarter than this. - So2

I don’t really like it because I feel like my code should work the first time and it never does, really. I mean I write it, and then I expect what I wrote to work, and it doesn’t, then I have to commit more time into this. - So2

Programming is debugging

I thought that’s what programming is? Is just fixing your errors. You just type it, you fix the errors, and you’re done. So, that’s all programming is. I think it’s like the same thing. Without it, computers would program themselves. I don’t know. I just think it’s the same thing. And if it worked on its first try, I really didn’t do much. - F6

People are careless programmers

Sometimes I don’t care. I just want to write stuff down and press go. - F6

Before in my head I have it worked out how its going to work, and then I glance and make sure its about what I want it to be. But, I usually just press go and check, and I don’t really look at what I wrote. - F6

Debugging is a way of learning

It would be better if I didn’t have to like fix anything, but then I guess I wouldn’t be learning anything. I don’t know. It’s just kind of part of it. Well, it just kind of seems essential because if I do it wrong, then of course it won’t work. - F3

I feel like debugging in itself and finding my own errors, I learn from those, and I’m less likely to make them [the same errors]. I enjoy it because it helps me learn to be a better programmer later on. - J1

Debugging helps me understand a lot better what other people have done. - P2

Miscellaneous

I feel really confident doing that. I mean, I’m the one who put them in there, so who else is going to find them? - F6

I think gaining some skill with it has made it enjoyable for me. - P1

Techniques people described as powerful

Just kind of knowing the power of your debugging is, I find that’s helpful - P3

It does always go back to, “What is the healthy state?” “What is the error state?” Those are very powerful control points. So either, if you you know you’re trying to resolve some error or reproduce an error, and then, so once you’ve got a hold of those... If I can a hold of those two points in the code, I can solve it. The points where even today I think at [his company] where we have bugs today, and we can’t solve errors, it’s because, in a production environment, we know that an error is occurring, but we can’t reproduce it locally because we can’t actually get a handle on when an error occurs. So, the big goal is always trying to get a good handle on a healthy state and an error state, and once I get that, I can solve any problem, and I think other people can generally. - P1

It was always, you know, someone who knew the tool set who showed me. So, it was always useful to have a friend, someone to show me. - P1

It is useful to learn, yet, so often, I do see people who are, you know, newer to software engineering, and by newer, I often think of people who have recently graduated, like I said I feel like I’ve learned a lot out of there. You know? So, something I always would recommend to someone is to learn different techniques and tools and all the variety of errors, you know, variety of tools. So often people learn to do... to simply put print line statements in their code. I said I love log statements, but uh... so it’s useful to learn a variety of tools. So it is powerful as a software engineer and makes someone very capable and able to work independently and be able to be able to produce, solve complex code, and produce complex code bases. - P1

I use print statements as a last resort. If there were a good debugger for PHP, I would use it first, hands down. - P2 (not captured on audio)

Why people think debugging is difficult

I think it’s not that complicated when you have a system that everything... you know, where your program is on your local machine. It gets more complicated when calling into underlying layers on remote machines. And that’s the hardest I think, for me. - P3

When asked about writing unit tests before coding: I’ve seen it end up being actually a negative thing because they end up over designing it. They create a bunch of stub classes that are not needed, and then it overcomplicates the system where it doesn’t need to be - P3\

Other interesting comments

You don’t want to insult anyone either. Because it’s, “Of course I can debug”, but they might not know. - P2

I’ve heard several people say that you should only go into like debugging mode, like, only, like, really step through your program, once you have an idea of what is going wrong, and you’re just confirming that’s what’s happening. And that is definitely not what I am doing. So, I feel like I could improve in the area of knowing my code in general better, so I can guess into it what is going wrong before I actually figure it out by trial and error. - J1