Attitudes and Mindsets Required to Be a Good Computer Programmer

John Karuitha, PhD

February 27, 2025

1 Background

Learning to program isn't just about mastering syntax and technical concepts. It also requires developing the right attitudes and mindsets, especially as you navigate the challenges of coding. Here are some key points to keep in mind as you start your programming journey.

1.1 1. Embrace Mistakes and Learn from Them

Programming is much like learning a new language—mistakes are inevitable, especially at the beginning. And just as when learning to speak a language, you'll make a lot of errors before you start to get things right. Some errors will be small, but they can still take time to resolve.

- **Example**: You may forget to add a comma or close a bracket, and spend an hour looking for the mistake. This is normal in programming.
- **“This computer hates me” feeling**: Many beginners feel like the computer is working against them, but that’s not true! When a computer throws an error, it’s simply saying, “I don’t understand what you’re trying to tell me.” Unlike humans, computers can’t infer meaning from small mistakes. For example, to a computer, `"Victoria:"` is entirely different from `"Victoria"` because of the colon.
- **Key Tip**: Don’t get frustrated when errors happen. Every mistake is a learning opportunity, and figuring out why something doesn’t work will make you a stronger coder.

1.2 2. Recognize Programming as Learning a Language

Learning programming is similar to learning a mother tongue. When you first start, you’ll make many mistakes and your “sentences” (programs) may not make much sense. The only way to improve is by constantly practicing.

- **Make Projects Early On**: The more you practice, the faster you’ll become fluent. Like learning any language, working on projects (even small ones) is key to building confidence and mastery. You might start by writing simple programs, like a calculator or a to-do list app.
- **Key Tip**: Think of programming as communicating with the computer. The more you “speak” through code, the better you’ll get.

1.3 3. You Don’t Need Deep Mathematics Knowledge

Many people mistakenly believe that programming requires a strong background in advanced mathematics. In reality, most programming tasks require only basic math skills.

1.3.1 What You Really Need:

- **Arithmetic**: Basic addition, subtraction, multiplication, and division.
- **Logic**: The ability to think systematically and break problems down into smaller parts.
- **Linear Algebra & Statistics**: While not always essential, basic knowledge can help in specific fields like data science or machine learning.

Key Tip: Don't let the fear of math discourage you from programming. In most cases, logical thinking and problem-solving are more important than advanced math skills.

1.4 4. Practice, Practice, Practice

Like any skill, programming improves with regular practice. The difference between good programmers and excellent programmers often comes down to how much they've practiced and applied their knowledge.

- **Code Every Day**: Even if it's for a short time, coding daily will help reinforce what you've learned.
- **Challenge Yourself**: As you progress, try solving more complex problems. Coding challenges and puzzles (like those on LeetCode, HackerRank) are great for sharpening your skills.

1.5 5. Stay Adaptable—The Programming Landscape Evolves

Programming languages and tools evolve constantly. While Python is very popular today, it may not be as dominant in the future. However, learning one programming language will help you pick up others more easily because most share similar concepts.

- **Transferable Skills**: Focus on mastering programming concepts (e.g., loops, functions, conditionals) rather than just the syntax of one language. These fundamentals apply across many languages, and once you know them, learning a new language is much easier.
- **Example**: If you know Python, switching to JavaScript or Ruby is much easier since they share common principles like loops and functions.

1.6 6. Understand the Precision Required in Programming

Unlike humans, computers are highly literal and get easily “confused” by small differences.

- **Attention to Detail**: A single misplaced comma or missing quotation mark can cause your program to fail. For example, `print("Hello, World!")` will work, but `print("Hello, World!` will throw an error because the quotation mark is missing.
- **Error Messages Are Your Friend**: When the computer gives you an error message, it’s not rejecting you—it’s telling you exactly what’s wrong. Learn to read and interpret these messages because they guide you to fix the issue.

1.7 7. Take Advantage of Abundant Online Resources

There’s no shortage of online resources to help you learn programming. Whether you’re looking for tutorials, documentation, or forums, you’ll find what you need on the internet.

1.7.1 Popular Online Resources:

- **YouTube**: Offers countless free tutorials for all levels.
- **Stack Overflow**: A community where you can ask questions and get answers from other programmers.
- **FreeCodeCamp**: Provides free coding tutorials and projects to practice.
- **Official Documentation**: Always refer to the official documentation of any programming language for detailed explanations and examples.
- **Large Language Models (LLMs)**: large language models like chatGPT can be of use when you are stuck. However, to master programming, try to first solve the problem by yourself and use LLMs as the very last resort.

Key Tip: When you encounter an error or get stuck, search online. Chances are someone else has faced the same issue, and you’ll likely find a solution.

1.8 8. Develop Problem-Solving Skills

Programming is about solving problems efficiently and creatively. You'll often need to approach problems methodically and think critically.

- **Break Down Problems**: Complex problems can be overwhelming. Break them into smaller tasks that are easier to tackle.
- **Be Patient**: Solving problems can take time. Don't rush—take a deep breath and approach each step thoughtfully.

1.9 9. Join a Community

Programming can feel isolating at times, but it doesn't have to be. There are thriving communities of developers, both online and offline, where you can seek help, learn from others, and stay motivated.

- **Collaborate and Share**: Engaging with other programmers, contributing to open-source projects, or even just asking questions will accelerate your learning.
- **Stay Motivated**: Being part of a group can help keep you on track when things get tough.

1.10 10. Cultivate Curiosity and a Desire to Learn

Programming is a field that rewards curiosity and a drive to keep learning.

- **Explore**: Don't just stick to what's familiar. Experiment with different languages, frameworks, and tools.
- **Ask Questions**: Never hesitate to ask "why" or "how." Understanding the underlying principles of how things work will deepen your knowledge.
- **Stay Updated**: The tech world moves quickly. Keeping an eye on new trends, technologies, and best practices will help you stay relevant (McKinney 2022).

---

2 Conclusion

Learning programming is an exciting journey filled with challenges and rewards. By embracing mistakes, practicing regularly, and adopting the right mindset, you'll not only become a competent programmer but also develop the resilience needed to thrive in an ever-evolving field. Remember, everyone starts as a beginner—what sets the best apart is their persistence and curiosity (Downey 2024).

References

Downey, Allen B. 2024. *Think Python.* 3rd ed. Green Tea Press.

McKinney, Wes. 2022. *Python for Data Analysis.* " O'Reilly Media, Inc.".