

Mentors Without Borders - Scientific Computing in Julia

Aryan Ritwajeet Jha

FAQ for Scientific Computing in Julia

1 What is Scientific Computing?

Scientific computing refers to the use of advanced computing capabilities to understand and solve complex scientific problems. It involves mathematical modeling, numerical analysis, and the use of computers to analyze and simulate scientific data. Basically, it is like regular programming, but with a higher emphasis on performance.

2 Why Julia for Scientific Computing?

Julia is a high-level, high-performance programming language designed for technical computing. It combines the ease of use of Python or R with the speed of C++, making it ideal for tasks in scientific computing. Key features include:

- High performance, approaching that of statically-typed languages like C.
- Dynamic typing and interactive use, similar to Python or MATLAB.
- An extensive mathematical function library.
- Very easy to setup on any PC, with a single install .

3 What is the intended outcome from this ‘course’?

- Practical programming experience for my students: To encourage my students to develop analytical skills to dissect a given problem at hand, and programatically implement solutions of their own to it.
- For my students to walk into a technical interview, or apply to a higher studies programme with confidence in their fundamental programming knowledge.

4 What is this ‘course’?

This course is based on Think Julia - How to Think Like a Computer Scientist by Ben Lauwens, a great book for beginners and experienced programmers alike. This book is available online for FREE, although a hard copy may be purchased if desired (not required for the course). Students will be assigned exercise problems from this book, which are meant to expose them to various data structures, and the nuances of using them when dealing with tradeoffs between speed, scalability and implementation complexity.

5 How will this course be delivered?

Contact Hours: One hour per week via Zoom, Saturdays 12 Noon Pacific Time, which are meant to entertain doubts from the exercise problems, or any other troubleshooting.

Correspondence: Slack Channels for any discussion throughout the week.

In this course, I do NOT:

- Deliver lectures

In this course, I do:

- Entertain doubts regarding exercise problems, or any programming problems in general

- Point students to other excellent resources available online for free, if they're interested in lecture based courses.
- Take feedback from my students regarding their current progress, and if they're having any difficulties, set up additional sessions to troubleshoot them
- Provide encouragement and guidance regarding careers in tech to students interested in the same

6 Who Should Take This Course?

Normally, studying scientific computing requires *some* prior programming experience. However, I've designed the course such that even first time programmers can This course is designed for:

- If you wanna break into tech.
- If you're already in tech but are uncertain about any of these terms: data science, data analysis, artificial intelligence, machine learning, data structure and algorithms, etc. : the stuff covered in this course will enable you to branch out in any of these directions
- You wish to pursue a higher degree in STEM

7 Who may Avoid This Course?

The course might not be the best fit for you if:

- You want to do only frontend development work such as web development, app development, etc. Making webpages or applications is not an intended outcome of this course.

8 Are There Prerequisites for This Course?

Knowledge: High School Maths including Algebra, Word Problems, knowledge of Decimals and Fractions, etc.

Infrastructure: A Computer with Internet Access.