**RYBEON**

Writingly and vocally

**CHAPTER 1**

**WHAT IS RYBEON**

The major programming suites (suites of technologies for creating softwares) we have are C, Java, C++, Python, Golang, Rust, and Ruby — none of these suite has all the 3 characteristics of an excellent programming suite (fulfilling, easy, and appealing).

If there is a programming suite that has all those 3 characteristics, won’t you choose it over those 6 major suites? If yes, then meet Rybeon.

Rybeon is a programming suite that has all the 3 characteristics of an excellent programming suite.

**Fulfillment**

**Ease**

**Appeal**

**CHAPTER 2**

**COMPOSITION**

Rybeon is composed primarily of — (i) an abstract programming language (Rybe) and (ii) a programmer’s code processor (PCP). There are additional, but secondary, useful components.

**CHAPTER 3**

**USER PERSONA**

Rybeon is designed to be suitable for all types of software engineers: governments, organizations, small businesses, groups, individuals, aliens, etc; this is contrast to mainstream GPTs which focus majorly on just individual software engineers.

**CHAPTER 4**

**GENESIS**

The founder of Rybeon (Brian Ibrahim Qamardeen) is a Software Engineer who had used a total of 12 programming languages (C, PHP, JavaScript, SQL, C++, Java, Python, Ruby, Lua, Kotlin, Golang, and Rust).

Often, when he picked up a new programming language, the language taught him new programming philosophies, some of which he found phenomenal and epiphanic.

After picking up Golang and Rust and consequently experiencing epiphanies again, he started to wonder if there were more beautiful programming philosophies. He wondered, “If there are more of them, how can I possibly acquire them all, instead of depending on accidents to discover them?”

After doing some thinking, he realized, “If I am to acquire all these beautiful programming philosophies that possibly exist, I have to start approaching programming according to how it should be approached, not how existing languages encourage it to be approached.”

When Brian started to approach programming according to how it should be approached (by repeatedly asking “How should programming be done?” and trying to find answers to it), he started discovering new philosophies (seeing ways programming could be better).

Brian started trying to apply his newly discovered programming philosophies to his day-to-day programming. Because existing languages were not designed based on these newly discovered philosophies, it was very difficult to apply them, using any of the 6 APTs.

As Brian could not apply his newly discovered philosophies, he decided to create a language that will enable him to do so. This decision gave birth to Operation Grusc (in October 2019), consequently birthing Rybeon.

**The Initial Idea Grew**

Although Brian’s initial intention was just to create a programming language he could use to apply his philosophies, he quickly realized that he had to morph the project into a bigger idea: a suite of programming technologies (containing (i) an abstract programming language, (ii) a programmer’s code processor, and (iii) some secondary supporting tools like - a reapplicable code infrastructure, a reapplicable codes collection, a codebase migration tool collection, etc). So Rybeon is now a suite of programming technologies, and not just a programming language. The language within it is instead called Rybe.