

GRQ00 - Programming Rust - Ch 1, 2

Your Name Here

Chapter 1. Why Rust

1. *Rust is generally a type-safe language. What benefits does type safety provide? How does type safety relate to static type checking in languages like Java, if at all?*

Your answer here. . .

Chapter 2. A Tour of Rust

Please read through the section “Handling Command-Line Arguments” and follow along with the exercises on your VM. Note, you do not need to follow the download and install instructions because Rust is already installed on your VM. If you have time and interest, feel free to read the rest of the chapter but know we do not expect you to know or understand this material.

2. *In the section “A Simple Function”, the first line of the `gcd` function invokes the `assert!` macro. What purpose does this line serve in the program? What is the purpose of using assertions in programs, generally?*

Your answer here. . .

3. *The `gcd` function returns the greatest common denominator of two unsigned 64-bit integers, but the function has no `return` keyword in it. Why not and how does Rust know what to return? When would you use the `return` keyword in Rust?*

Your answer here. . .

4. *Rust does not have exceptions. Instead, functions either panic, and crash the program, or return a `Result` value. Explain what a `Result` value is in your own words.*

Your answer here. . .

5. *Consider a language you know well. Ignoring the differences mentioned above, in these early code examples what stands out to you as unique to Rust versus another language you know?*

Your answer here. . .