Assignment: Exception Handling Across Multiple Programming Languages

1. An exception is essentially an error, it is a special condition that disrupts the normal flow of a program, often to invalid inputs or operations. Java relies on try-catch blocks for exception handling, which is fairly robust and has plenty of built in exceptions. C has no built in exception handling, it must be managed manually using mostly conditional checks. Go returns error values from functions, mixing explicit error handling control with simplicity. Scheme can only use conditional expressions to manage error, which makes sense considering the level of the language. Prolog also only can rely on conditionals like logical conditions to handle errors. And python provides try-except blocks that have some built in exceptions like java.
2. I have a personal bias towards java as the language I've learned the most, but python handles exceptions in a much more clean and intuitive way. Generally, all languages have some kind of benefit that could make them useful in some circumstances. Java is robust and has lots of capabilities, but it is fairly slow in terms of development compared to other languages, and it's more resource intensive. C is lightweight and allows absolute control over the exceptions, but it lacks many built in mechanisms leaving everything to user error. Go is explicit and reduces hidden bugs, but it lacks some of the customization of the others. Scheme and Prolog are both consistent and functional while using few resources, but lack much of the structured or intuitive systems that the other languages have. And python is flexible and highly readable, though it does take plenty of resources.
3. Implementing error handling in languages without built in systems, like C and Prolog, was particularly confusing. In C, the reliance on manual checks required meticulous care did interest me as I can see how it might be useful, though it felt unintuitive. In Prolog, using logical conditions to detect errors felt less natural too, giving that same sense of confusion when writing it.