

Building this project helped us gain a better understanding of Exception Handling via real implementation through Python, Ruby, and Java. One of the biggest things we learned was how to handle the exceptions in Ruby. Practicing in each language showed me how each language deals with errors and program flows. One of the main things we learned was how propagation works. In class, it's explained that an unhandled exception moves up the call chain until it reaches a handler, and if it reaches the main program without being handled, the program terminates. We were able to see this directly when we forced each program to open a file that didn't exist. The message printed in the main proved that the propagation was working exactly how the slides described. Another important part of Exception Handling that we tackled is the difference between exception models in each language. Python was the simplest because it doesn't have checked exceptions, using try, except, and finally structure matches the examples in class. Ruby we had the least experience in, but also followed the rescue and ensure, so it was easy to compare it to Python's version. Java was the most unique, it forces the programmer to acknowledge which exceptions might happen, something the slides described when talking about checked vs unchecked exceptions. Overall, this project helped us connect the lectures in class and provided real-life practice. Now we understand exception handling in multiple languages and how different and similar it can be across Python, Java, and Ruby.