Isaac Bleecker

Adapting to Different Languages

Figuring out C++'s syntax made it one of the more difficult languages I have tried to learn. However, the logic behind programming fundamentally stays the same from language to language, and C++ has many similarities to Java. For me, the best way to learn the language while working on this project was to attempt to write a line, compile it, and deal with any error messages. My strategy essentially became rapid trial and error throughout the project, as I knew I would make a lot of little mistakes. Whenever I wrote a larger block of code, I would compile after every few lines. That helped me hone in on bugs as they came up.

If the line I wrote compiled without problems, then I didn't have to look up anything. However, when I inevitably ran into a variety of errors concerning semicolons, types, passing parameters, calling functions, balancing brackets, or declaring objects and variables. For each error, I would look back at my code and try to spot any obvious issues. If I couldn't find the specific problem to fix, I would google search the bug message, and try to find a source that helped with the issue. Error messages alone are not always the most helpful guidance in fixing a bug, as they sometimes identify where the bug caused problems, rather than where and why the bug actually exists. I did not use any consistent online source for that, I would just choose a top result on Google that appeared to translate the error message into more easily understandable terms.

Another resource I found myself using a lot was the official C++ documentation, mostly to figure out how certain operations or objects were supposed to be used. For example, I referenced the official documentation on enumerations frequently.