

Zac Foteff

Dr. Bowers

CPSC 326: 01

20 April 2021

### Final Project Progress Report

Having worked on this project for some time now I have a good amount of the project done, but I still have a fair amount of work ahead to get this project into a workable state. To implement these try-catch blocks into the MyPL language, I started by adding the tokens TRY, THROW, and CATCH into the list of acceptable tokens to MyPL. An important correction I need to make to my project specification is that I confused the terms 'throw' and 'raise' as a result of trying to decide which term I wanted to use for the token corresponding to a user triggered exception state. The final term I included into this version of MyPL is 'throw' and will be used from here on. I then modified the Lexer file to output the TRY, THROW, and CATCH tokens on their appearance in the source file. Then I moved on to adding the TryStmt and CatchStmt nodes into the AST for MyPL. I chose not to include another node for throw, since it should only be an id token or an expression which be stored in the TryStmt AST node. In the Parser, I added the proper functions for building TryStmt and CatchStmt nodes. Additionally, I modified the pretty printer to handle try-catch blocks to ensure the nodes are added properly to the AST. Moving forward, I still need to add the grammars to the Type checker, and I need to implement the try-catch blocks in MyPL with C++ try catch blocks. I don't have a properly working Interpreter, which is impeding a lot of my progress. I will ensure with the time left that I have a properly working implementation of MyPL that implements these try-catch blocks by conditionally adding and removing environments on the stack. I believe at some point in my proposal I included break statements as an additional example to the exception handling; I will have to reduce the scope of my project to not include break statements, as the try-catch blocks are already providing me a significant enough challenge and break statements may be an overwhelming addition to handle.

To run this work in progress version use: `$ make`, then `$ ./mypl tests/<test>` with one of the included tests. This will use the pretty printer to show the proper implementation of the try-catch blocks into the AST.