Weston Averill

CPSC326

5/7/2021

Final Project Report

Extension: The goal was to implement the equivalent of C++ pointers

What was accomplished: The first thing that I was able to accomplish was being able to declare pointers. You can do this either by using “var ~ID : int = nil” or “var ~ID = &ID2”. The types of the pointers are assigned the same way as a normal variable in MyPL. With my pointers, if the pointer value is changed, the referenced variable will also be changed. I can do all math operations with pointers and conditional statements. I was also able to pass my pointers to functions. This means the pointers were able to be modified in other functions.

What wasn’t accomplished: I had a difficult time trying to figure out how I would be able to use the memory address when referencing variables. Instead, I had to figure out a way to update the pointers values when the variable it references was modified.

Testing: First I had to test to make sure that when I created my pointer variable, that it would be able to grab the value of the variable it was referencing. After that, I tested all of the math operations with the pointer variables. I was testing the math operations with all of the different data types as well. I also had to make sure that when a pointer changed, the referenced variable had to be modified and vice versa. Then I tested just simple conditional statements. Finally, I tested my pointers by passing them to functions. I had to make sure that they could be modified in the functions and keep that value after returning from those functions.

Building/Running: After pulling the source code from GitHub, just run the cmake command with CMakeLists.txt to build the executable. Use the command “./mypl tests/basic-pointers.mypl” to test the pointer extension I implemented.