CPSC 326: Homework 0 *Due By Mon., Jan. 22*

Overview. The goal of this assignment is to ensure that you have your environment set up for completing and submitting homework assignments 1–6 (MyPL code development).

Step 1. Log into piazza (instructions should be emailed to you from piazza) and post a question you have regarding MyPL (syntax, features, etc.) under the `mypl` folder.

Step 2. Install python (version 3.10.12 or higher) and pytest (version 7.4.4 or higher) on your local machine if you don't already have them installed. Ensure you can run both python and pytest from the command line via a terminal shell.

Step 3. Obtain the starter code from GitHub classroom. Details will be provided on piazza. The starter code comes with some unit tests in the file `hw0_tests.py`, a file `mypl_error.py` that contains a single class (for representing MyPL exceptions/errors) and corresponding helper functions, a file `mypl.py` that contains a "driver" program for MyPL, and a bash script called `mypl` that can be used to execute the driver program. Take a few minutes to look through each of these files.

Step 4. Use pytest to run the unit test file `hw0_tests.py` provided in the starter code. To run the tests, from the command line in the starter code directory type:

```
pytest hw0_tests.py
```

After running the command you should see that the 4 tests in the file each pass. Your output should look something like the following.

```
============================= test session starts ==============================
...
collected 4 items

hw0_tests.py ....                                                        [100%]

============================== 4 passed in 0.00s ===============================
```

Step 5. Use python to run the MyPL driver program. For help on using the program, you can run the following from the command line (again, from the same folder as the starter code).

```
python mypl.py -h
```

The result should be the following.

```
usage: mypl [-h] [--lex | --parse | --print | --check | --ir] [filename]

Run the mypl interpreter. If filename missing, reads from standard input.

positional arguments:
  filename    mypl program file (optional)

options:
  -h, --help  show this help message and exit
  --lex       displays token information
  --parse     checks for syntax errors
  --print     pretty prints program
  --check     checks for static analysis errors
  --ir        displays intermediate code
```

Note that if you are using Linux, MacOS, or a Bash shell (e.g., in VSCode through a Git Bash shell), you can run the above via the simple `mypl` bash script provided in starter code as follows.

```
./mypl -h
```

This is meant to be a more convenient way to run the `mypl` interpreter in later assignments.

Step 6. Create a document that shows your results of trying the different options listed above (e.g., `python mypl.py --lex`, and so on), including the use of no options. In your tests, also experiment with providing both an existing and non-existing file (which would presumably be a MyPL source file). Note that for each case, you should get some type of error message. Save your document as a PDF file and name it `hw0.pdf`. Submit your file as part of your repository (i.e., add, commit, and push it). Double check on the GitHub page for your project that the file was successfully submitted.