

Programming Assignment 1

P1.java

This program tests all of the Sym and SymTable operations.

It should only print the following text if there is no error.

```
1  === Sym Table ===
2
3  {}
```

Additional output is only produced if one of the methods that it is testing does not work as expected

Assumption: `Sym.java`, `SymTable.java`, `WrongArgumentException.java`, `DuplicateSymException.java` and `EmptySymTableException.java` are present in the same director as `P1.java`

Usage: `javac *.java`; `java P1`;

SymTable.java

The class SymTable is used to represent a symbol table

- **Sym(String type)** This is the constructor; it initializes the Sym to have the given type.
- **String getType()** Return this Sym's type.
- **String toString()** Return this Sym's type.

Sym.java

This class represents a symbol in a symbol table

- **SymTable()** This is the constructor; it initializes the SymTable's List field to contain a single, empty HashMap.
- **void addDecl(String idName, Sym sym)** If this SymTable's list is empty, throw an EmptySymTableException. If either idName or sym (or both) is null, throw a WrongArgumentException. If the first HashMap in the list already contains the given id name as a key, throw a DuplicateSymException. Otherwise, add the given idName and sym to the first HashMap in the list.
- **void addScope()** Add a new, empty HashMap to the front of the list.
- **Sym lookupLocal(String idName)** Sym lookupLocal(String idName)
- **Sym lookupGlobal(String idName)** If this SymTable's list is empty, throw an

EmptySymTableException. If any HashMap in the list contains idName as a key, return the first associated Sym (i.e., the one from the HashMap that is closest to the front of the list); otherwise, return null.

- **Sym lookupGlobal(String idName)** If this SymTable's list is empty, throw an EmptySymTableException. If any HashMap in the list contains idName as a key, return the first associated Sym (i.e., the one from the HashMap that is closest to the front of the list); otherwise, return null.
- **void removeScope()** If this SymTable's list is empty, throw an EmptySymTableException; otherwise, remove the HashMap from the front of the list. To clarify, throw an exception only if before attempting to remove, the list is empty (i.e. there are no HashMaps to remove).
- **void print()** This method is for debugging. First, print "\n= Sym Table =\n". Then, for each HashMap M in the list, print M.toString() followed by a newline. Finally, print one more newline. All output should go to System.out.

WrongArgumentException

This exception is thrown when there is something wrong with the argument

EmptySymTableException

This exception is thrown when then symbol table is empty

DuplicateSymException

This exception is thrown when there is a duplicate symbol added to the symbol table