

## Usage:

- `javac -d ./bytecode *.java && (cd ./bytecode && java P1)`

## Prerequisites:

- OpenJDK 11

## Tests documentation:

### **public class P1**

P1 class: tests the SymTable & related classes - Sym

Note: test methods tend to preserve names of corresponding methods being verified & are prefixed with “test\_”.

Note: “print” method is used althroughout the unit tests, which excludes it from being tested.

### **public static void main(String[] args)**

test driver: calls unit test methods & annotates each test with pass/fail status

- Parameters: `args` — input arguments if any

### **public static boolean test\_Sym()**

checks the correctness of the constructor & related getter/setter methods implemented in the Sym class

### **public static boolean test\_exception()**

checks the correctness of custom checked exception classes

### **public static boolean test\_SymTable()**

checks the correctness of the constructor implemented in the SymTable class

### **public static boolean test\_addDecl()**

checks the correctness of the addDecl method in the SymTable class

depends on removeScope method

**public static boolean test\_addScope()**

checks the correctness of the addScope implemented in the SymTable class

**public static boolean test\_lookupLocal()**

checks the correctness of the lookupLocal implemented in the SymTable class

**public static boolean test\_lookupGlobal()**

checks the correctness of the lookupGlobal implemented in the SymTable class

**public static boolean test\_removeScope()**

checks the correctness of the removeScope implemented in the SymTable class

---

## Compiler for a C- language

- C- is a simple programming language that uses Pascal identifiers.
- Assumption: the compiler lexical and syntax analysis phases would be executed in parallel, where the scanner would stream tokens to the parser for evaluation. The compiler would insert & remove current scopes from the list while the analysis process is being executed, and priority is given to the current scope (latest pushed to the list).

### Terminology:

- identifier: lexical token naming language entities (e.g. variable or function names).
- symbol: memory instance storing information about a corresponding identifier token.

---

### To-do list:

#### Assignment submission:

- Create pdf from markdown: `pandoc README.md -o <lastname.firstname.Pn.pdf>`
  - generate markdown from javadoc and remove redundant comments or
  - generate javadoc to extract method headers: `find . -type f -name "*.java" | xargs javadoc -d ../javadoc`
- Add headers for each file
- Verify code format
- Verify code execution on CSL machines