

## Usage:

- `make clean && make testclean && make`
- `make test`
- `diff testErr.expected testErr.out`
- `diff test.expected test.out`

Executing the test will generate output file of the formatted program.

## Prerequisites & dependencies:

- OpenJDK 11
- Java Cup
- JLex

## Tests documentation:

- use the Linux utility `diff` to compare your file of error messages with the expected files.
- 

## Type checking for C—:

The type checker will determine the type of every expression represented in the abstract-syntax tree and will use that information to identify type errors.

## Types:

- primitive types: `int`, `bool`, `string`, `void` (*function return type only*)
- Type constructors: `struct` (*includes name*), `functions` (*includes params & return types*)

## Operators:

- logical: `not`, `and`, `or`
- arithmetic: `plus`, `minus`, `times`, `divide`, `unary minus`
- equality: `equals`, `not equals`
- relational: `less than (<)`, `greater than (>)`, `less than or equals (<=)`, `greater than or equals (>=)`
- assignment: `assign`

## Rules:

- [x] Limited coercion (no implicit type conversion): bool cannot be used as int, & vice-versa.
- Type errors:
  - [x] Arithmetic operators: int operands; int result;
  - [x] Relational operators: int operands; bool result;
  - [x] Equality operators (==, !=): same type operands of int, bool; bool result;
    - \* (and cannot be applied to function names, struct names or variables.)
    - \* NOTE: don't need to worry about equality operators between string literals.
  - [x] Assignment operator (=): same type operands (RHS & LHS) of int or bool; result in RHS type;
    - \* (and cannot be applied to function names, struct names or variables.)
  - [x] logical operators & conditions (if, while) must have bool operands.
  - [x] input/output operators (cin >> x, cout << x):
    - \* cout: int, bool, string literals operands;
    - \* cin: int, bool identifier operands;
      - NOTE: can be int/bool field accessed using dot operator;
    - \* x cannot be function name, struct name or variable.
  - [x] function invocation: call on function identifier; matching args (formals = actuals) types & number;
  - [x] function returns: must return value of correct type for non-void function;
    - \* void function must not have a return statement (even without value).
  - [x] repeat loop times clause: int operand.

## Tasks

- [x] check example files.
- [x] Write test files.
- [x] Edit P5.java, ast.java.
- [x] Type checking implementation:
  - add typeCheck method to AST nodes - walk the AST checking types of sub-expressions.
  - for base nodes (e.g. IntLitNode) just return their type.
  - for IdNode: lookup the type of the declaration in the symbol table (linked field), and propagate up the type.
  - Algorithm: get LHS & RHS types, check types compatibility for operator, then set the kind of node to be a value, then set type of node to be the type of the operation's result.

- [x] function calls: get type of each actual argument, match against the formal argument (check symbol), propagate the return type.
- [x] Cascading errors: to prevent duplicate error reporting, introduce an internal error type (use a special `ErrorType` for expressions that contain type errors):
  - when type incompatibility is discovered then report the error, and propagate the type up the recursive calling.
  - when an error type is encountered as an operand: don't re-report an error, only propagate the error type up the calling chain.

### 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
- [ ] lastname.firstname.lastname.firstname.P5.zip +--+ deps/ +--+ ast.java +--+ cminusminus.cup +--+ cminusminus.jlex +--+ DuplicateSymException.java +--+ EmptySymTableException.java +--+ ErrMsg.java +--+ Makefile +--+ P5.java +--+ Sym.java +--+ SymTable.java +--+ Type.java +--+ typeErrors.cminusminus +--+ test.cminusminus +--+ lastname.firstname.lastname.firstname.P5.pdf