ToyC Compiler

EGRE 591 - Dr. Dan Resler

Samuel Coppedge & Xander Will

Fall Semester 2019

How to Use

To use the ToyC Compiler, you must first have installed Java and Apache Ant, and have those running successfully on your environment.

To begin, `cd` to the /compilers/toycJAVA directory. In the directory labeled "tests" you can store you .tc source code files and run them from the parent directory. There are a handful of testing files in the tests directory for trying out given language capabilities if you choose to use those.

To run your .tc source code file, you'll need to be toycJAVA directory and run ar `ant compile` command. An `ant clean` may be necessary beforehand if you have changed some of the code and the program is not compiling correctly.

The main command to run your code will be in the form of`ant -Dsource="<FILENAME>" -Dflags="<OPTIONAL FLAGS>" test`, where \ is the name of your .tc source code file in tests (the .tc in the command is not necessary) and where \ is a combination of 0 or more of the compiler flags included in the Toy C specifications.

Successfully compiled .tc code will output a .j file in the directory. By default, this will be the same name as the .tc source code file you compiled. To run your code, input `java -jar jasmin.jar <FILENAME>.j` to compile your jasmin file into a Java .class file, then `java <FILENAME>`.

Two helpful batch files have been included in the toycJAVA directory to streamline this compilation and running process. If you are running on Windows, the command `./tc.bat <FILENAME> "<OPTIONAL FLAGS>"` will compile your code, and a `./tcrun.bat <FILENAME>` will run it. Note that the\ should be put in quotes if there is more than 1, and that the \ doesn't require either `.tc` or `.j` for either command.

Optional Compiler Flags

According to the ToyC specifications, the compiler flags are as follows:

```
 -help:                  display a usage message
-debug  <level>         display messages that aid in tracing the compilation process
-output <file>          specifies target file name
-class  <file>          specifies class file name

<level>:
        0 - All messages
        1 - Scanner messages only
        2 - Parser messages only
        3 - Code generation messages only

-abstract               dump the abstract symbol tree
-symbol                 dump the symbol table(s)
-code                   dump the generated code
-verbose or -v          display all information
-version                display the program version
```

Please note that due to the ToyC compiler being written using Java, if the `-output` and the `-class` flags are present without one another or do not match the same string, you code will not run.

Noteable Features & Quirks

- #### Function calls and recursion
- #### Loops and Conditionals
- #### Multiple assignment (works in conditionals)
- #### Reading user input and Writing to the console
- #### Global variables and local scoping
- #### Slight code optimization

Please note that:

- #### Variables must be declared at the top of their scope and cannot be redeclared
- #### `char` as a data type is restricted despite being recognized by the language