

# CS126: Introduction to Compilers

## Assignment 1: SAMPLE

**Out: 9/5/03**

**Due: 9/12/03**

### 1.0 Objective

The purpose of this assignment is three fold. The first is to let you become familiar with the DECAF programming language for which we will be writing a compiler. The second is to become familiar with the skeletal compiler code that is being provided to the class, including the coding conventions, Java style, and overall structure. The third is to build up a repertoire of test programs that the whole class can use later to test their compilers.

### 2.0 Sample Program

Your first task is to write a sample DECAF application. It doesn't have to be that long (50-200 lines is probably about right, but you can make it longer if you want to). You should read enough of the handouts to understand the language (which should be easy since its pretty much a subset of Java which you should already know). Then, come up with a program idea and implement it. Try to avoid duplicating what is already present in the tests directory and (as much as you can) what other students are doing.

You can run the current experimental DECAF compiler to check your program and see if it works. (Note that the compiler is relatively new and not necessarily completely debugged. Report any problems to [spr@cs.brown.edu](mailto:spr@cs.brown.edu).)

You should hand in a tarball or zip/jar file containing:

- The final working source code. The code should have enough comments in it so that the other members of the class can determine what the program does, what the expected inputs are and what the expected output should be. Some sample input/output pairs would also be helpful.
- All versions of the program that you created that had syntactic or semantic errors or caused the experimental compiler to abort (and thus wouldn't compile for some reason). These will be helpful for testing error recovery and error detection in the compilers you are going to write.

## 3.0 Code Reading

You should also read the code in the top-level directory of the DECAF compiler. For each class/interface listed below in this directory, write up a brief description of what you think the class does. Include a description of any methods and what you think they are used for.

The relevant classes at this point are:

```
DecafAst.java
DecafBuiltin.java
DecafConstants.java
DecafErrorHandler.java
DecafException.java
DecafFactory.java
DecafMain.java
DecafModifier.java
DecafParser.java
DecafScope.java
DecafSymProcess.java
DecafSymbol.java
DecafType.java
DecafTypeProcess.java
DecafVisitor.java
```

Also, look at the coding style and write up a brief description that lists the naming and other stylistic conventions that are used throughout the compiler.

## 4.0 Mechanics

Handins should be done electronically. (Directions will be posted in the cs126 news group.) You are free to work in small teams on this assignment (and future ones as well). On this assignment, if you are working in a team, I would like the team to hand in a separate sample program for each team member. You should also make sure that each team member understands the compiler organization and coding styles.