COP5555 Fall 2013
Assignment 5

Assigned Nov 2
Due: Wed Nov 13 at 11am

The attachment contains a class Compiler, and an incomplete implementation of CodeGenVisitor. It is complete, enough, however, to run the little program in the file show_cise_image.plp. Your assignment is to fill in parts of the implementation of CodeGenVisior so that the compiler can handle a wider range of programs. Assignment 6 will add the remaining functionality. What you need to complete is described in the CodeGeneration attachment.

1. Get your environment set up so that you can build the Compiler class, use it to compile show_cise_image.plp. To build the Compiler, you will need to get asm-4.2 from http://asm.ow2.org/ .
2. Execute the classfile you obtained from step 1. We are using classes in the cop5555fa13.runtime package to support execution so the classfiles from this package are needed to run the classfiles obtained by compiling our language; their location should be specified in the classpath. Even if you are using an IDE, you might find it easier to execute your generated classfile using "java" on the command line and explicitly indicate the location of the cop5555fa13.runtime with the –classpath option.
3. Add functionality to the CodeGenVisitor to handle more of the language. See the attachment for more details. Work incrementally, testing as you go. For the most part, you will implement straight line programs, leaving loops and branches for assignment 6.
4. Do not change the given classes. During grading, they will be replaced with another version that emits generated images and other data to allow comparison with items generated by a reference compiler. If you want to add additional methods or fields, create a subclass.

Hint: An ASMifier tool is useful—it takes a classfile and shows the asm calls that would be used to generate it. The ByteCode Outline plugin from http://andrei.gmxhome.de/bytecode/ provides eclipse support. Once the plugin is installed, go to Window→Show view→Java→Bytecode. This will show the bytecode of the java program open in your editor. The ASM button toggles between showing the bytecode and the ASM calls that would generate it. The button with two horizontal arrows toggles whether it is linked to the arrow or not. You can explore the remaining menu items yourself. If you don't use eclipse, look for the ASMifier class in the org.objectweb.asm.util package that comes with the asm distribution. It contains a main method and takes a classfile and prints the asm calls that would be needed to generate it.

Submit to elearning:

One jar file called Assignment5.jar containing the source code of ALL class (including those provided by me) in your program.