Homework 1 – Profiling in OpenJava and Javassist
CS 420/520

Assigned: February 1, 2021
Due: February 17, 2021

Honors Statement

The work that you submit must be your own. You should not receive detailed help from any other student regarding this assignment, and you should not offer your solution to another student. Nor should you go out to a newsgroup and ask for help from the global community. You should not ask questions about the assignment to other students – all questions should be addressed to Jeff for clarification purposes.

How to Submit

You must submit a SINGLE zip file on Blackboard. This filename should be named by your last name (lastname.zip). This zip file must contain two other zip files (p1.zip and p2.zip) that contain your solution files, as stated in the problem description.

The assignment must be submitted by midnight (Apia time -ha!) on the due date and follows the late policy from the syllabus.

Common Files Needed

The OpenJava and Javassist tools are zipped and available on the Google Drive. We will do live examples over the next few lectures so you can understand how to use these tools.

You will also be provided with the code that you must instrument. The code will contain several simple base classes. Your job is to write the metaobjects that will adapt each class to the described profiling capability. These files will be used to grade your submission.

Problem 1 - Simple Profiling using OpenJava

In this problem, you must write OpenJava code that will simulate the idea of profiling. The idea will be to instrument several base classes with the code that will trace the entrance and exit of a method, along with the amount of time spent in each method. You will likely need to create other class(es) to support the tracking of the internal clock time and the instrumentation of the sample code with this new profiling feature.

Your submission for this problem should contain a file called p1.zip that includes the OpenJava file representing the metaobject that adds this behavior. If you need to modify the headers of the provided files, in order to attach metaobjects, also include those. Please include any other supporting classes or metaobjects that you need to complete the problem. You should submit everything needed for us to grade the problem contained in this p1.zip file.

The output from a successful project will resemble the following (only the output of the metaobjects is include below – there is also output from the base classes, but not included here):

```
Entering main…
Entering A::foo…
Entering A::bar…
Leaving A::bar. Time spent: <some time here>
Leaving A::foo. Time spent: <some time here>
Entering B::foo…
Entering B::bar…
Leaving B::bar. Time spent: <some time here>
Leaving B::foo. Time spent: <some time here>
Entering C::foo…
Entering C::bar…
Leaving C::bar. Time spent: <some time here>
Leaving C::foo. Time spent: <some time here>
Entering A::foo…
Entering A::bar…
Leaving A::bar. Time spent: <some time here>
Leaving A::foo. Time spent: <some time here>
Leaving main. Time spent: <some time here>
```

Problem 2 – Simple Profiling using Javassist

This problem is the same as Problem 2, except your solution must use Javassist.

Your submission should contain a zip file called p2.zip that includes the Java source code representing the solution using Javassist. You must use the same sample test files.

Additional Work for CS 520

The students in CS 522 must do additional work for this assignment. In addition to displaying the time in each individual call as shown above, the 520 students must give a final report of the cumulative time spent in each method call during the entire execution. For example, a successful 520 solution will not only provide the output shown above, but also produce the following output at the end of the execution:

```
<SAME OUTPUT AS CS 420>
Total time spent in each method call (cumulative):
A::Foo: <some time here>
A::Bar: <some time here>
B::Foo: <some time here>
B::Bar: <some time here>
C::Foo: <some time here>
C::Bar: <some time here>
main: <some time here>
```