# 17-654: Analysis of Software Artifacts

Jonathan Aldrich

Assignment 6: Hoare Logic and ESC/Java

Turn in a file named username-Assignment6.zip, where username is your Andrew id. The file should contain Stack.java, stack-output.txt, and answers.{txt,html,pdf} (with your answers to the questions). In a comment at the top of the answers file, state your name, Andrew id, and how long you spent on the assignment.

This is an individual assignment, except for section 3, design presentation preparation. For section 3, turn in a single file username1-DesignPresentation.{ppt,pdf} for the entire group.

**Assignment Objectives:**

- Write with specification constructs including pre- and post-conditions, loop invariants and variant functions
- Prove small programs correct using Hoare logic techniques
- Use ESC/Java to check functional correctness properties of programs

# 1 Hoare Logic (40 points)

Consider the following WHILE program:

$i := 1;$
$r := 1;$
$while\ (i < n)\ do$
  $i := i + 1;$
  $r := r + i$

**Question 1** (10 points).

For the while program given above, state a (a) precondition, (b) postcondition, (c) loop invariant, and (d) variant function. The postcondition should precisely define the value of $r$ in terms of $n$. To help you out, note that $\sum_{i=1}^{N} i = \frac{N*(N+1)}{2}$

**Question 2** (20 points).

Using weakest preconditions, state the proof obligations that assure (a) the invariant is initially true on entry to the loop, (b) the loop invariant holds after execution of each loop, (c) if the variant function reaches zero, the loop will exit, (d) the variant function decreases in the loop, and (e) the postcondition holds.

**Question 3** (10 points).

Prove each of the proof obligations above. Your proof should be done at the level of detail shown in lecture, i.e. small steps with justifications given for each step.

## 2 ESC/Java (60 points)

**Instructions.** Download ESC/Java 2. There are two options, the command-line tool, and an Eclipse plug-in.

**Command Line Checker.** You can download the command-line checking tool (we tested with binary version 2.0.5) at:

`http://secure.ucd.ie/products/opensource/ESCJava2/download.html`

Follow the instructions in README.release to install the system. Install ESC/Java in a directory path without a space; in my experience having a space in the directory path can cause setup problems.

The installation instructions for Windows are not as clear as one might like. If you are working from Windows, you will need to modify escj.bat as follows:

- Change the line "set ESCJAVA_ROOT=..." so that the right of the equals sign refers to the directory where escjava is installed (the directory where escj.bat is).

**Eclipse Plug-in.** Instructions for downloading and installing the Eclipse plugin are at:

`http://kind.ucd.ie/products/opensource/ESCJava2/escjava-eclipse/updates/escjavaHelp.html#Installation`

**Question 4** (50 points).

Now take the file Stack.java and StackCheck.java from the class website. Run ESC/Java 2 on both files together:

escj Stack.java StackCheck.java

Add pre- and post-conditions and invariants to Stack.java. You may find bugs in the code as you go along–if so, you may fix any unambiguous bugs that you find in Stack.java (and in fact, you may have to fix them in order to get ESC/Java to pass the code). When you are done, ESC/Java must run on both files without producing any warnings. You may not edit StackCheck.java. Nor may you remove the annotations that are already present in Stack.java.

**Turn in:** (a) your edited version of Stack.java, and (b) a printout of ESC/Java 2's output when run on the two files in stack-output.txt.

If you are using ESC/Java from within Eclipse, you will need to capture ESC/Java's output in the console window. To do this, first make sure ESC/Java is set up to give you detailed output. Go to Window — Preferences — Java — ESC/Java2 and UNCHECK the box "Disable ESC/Java informational messages". You should get lots of information when you run ESC/Java. To make sure you get a clean capture, before you run ESC/Java right-click on the console and select the Clear option.

A final hint: sometimes ESC/Java works better if you express invariants directly in terms of fields like topOfStack rather than in terms of calling other functions like isFull(). My guess is that this is just a limitation of the tool.

**Question 5** (10 points).

In the file answers.txt, criticize the ESC/Java tool (1-2 paragraphs). What did you like about it, and conversely what stands in the way of making this a practical tool?

## 3 Framework Design Presentation (30 points)

At Bored Games Software, word is that a bunch of upstarts have come up with an alternative to your carefully worked out framework design. Although management did not request the alternative, they are concerned about getting the design right and have scheduled a company meeting where both designs will be presented, after which a design selection committee will choose one of the framework designs to move forward with. So you need to prepare a short but compelling presentation describing your design.

Specifically, prepare an 8-minute presentation (+ 2 minutes of questions is 10 minutes total) to be given in class on the goals for your framework, your design, and the rationale behind that design. Your primary audience is your classmates, and your job is to convince them that they want to write a game or UI plugin for your framework. Your grade will be determined by the clarity of your design presentation and how well your design is justified. If you wish, you may include extra slides or other design materials for your classmates to view online at their option; these extra materials will not be graded.

To give you context: after presentations are given in class, groups will bid for tasks in assignment 8. Possible bids will include (a) working on your own framework; (b) if another group proposed a framework design that requires two teams, you may bid to be the second team on another framework implementation; (c) implementing a UI plugin for another team's framework (specify what UI platform you will target; and (d) implementing a game plugin for another team's framework (specify which game). Each team *must* bid for at least one UI plugin and one game plugin for another team's framework, but may specify a priority order.

Bids will be chosen to maximize overall team preferences, subject to the constraint that complete applications are formed, i.e. a framework, at least one UI plugin, and at least one game plugin. If you can convince other teams to bid to plug into your framework, you get the plum job of implementing your design, 10 points extra credit on this assignment, and of course, undying glory.