



Automating Feedback for 1st year Programming Courses

Gudmund Grov

Graeme Reid

Verena Rieser

Computer Science (MACS)



**Programming is a key
skill for a computer scientist**

50% of our first year CS
modules teach
programming



... but programming is a big hurdle for many 1st year CS students

~~I'm a Programmar~~
~~I'm a Programer~~
~~I'm a Programmor~~
I write code.

Studies indicate that potential issues are

constrained lab times

large classes

insufficient feedback



**Lab helpers are invaluable but
resources are limited...**

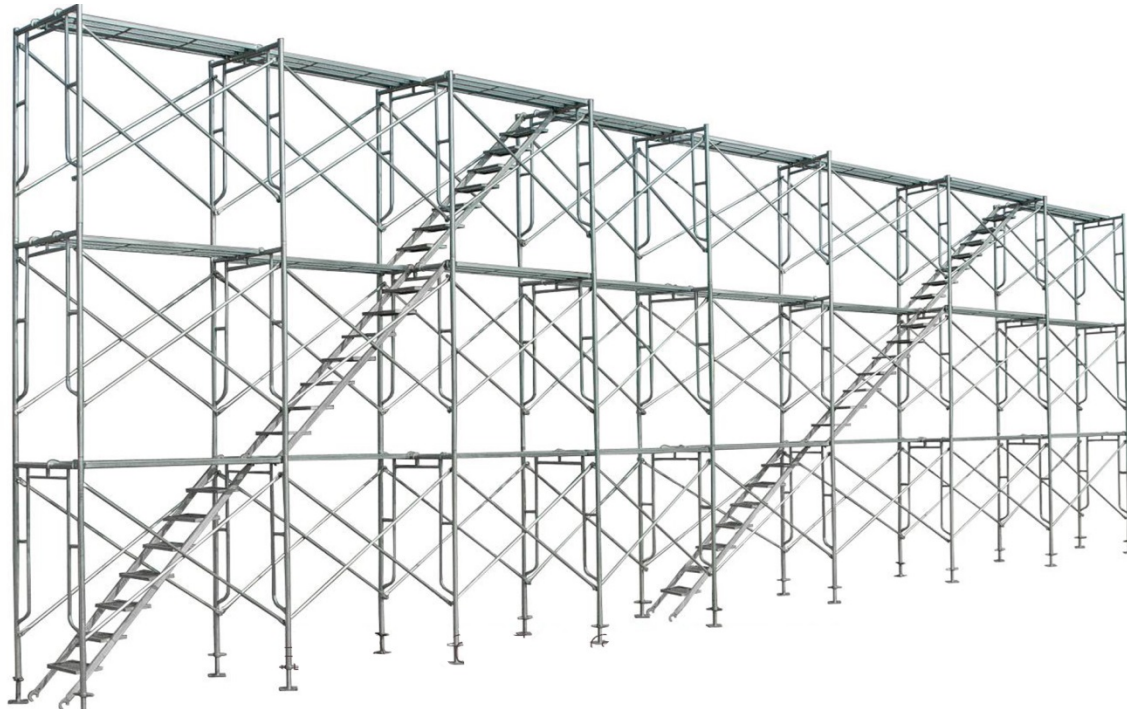




Develop **automated** feedback
framework that is **timely** &
constructive



Scaffolding: feedback in terms of clues & tips



Software testing to check results from running program





ASM library to analyse structure of running program



Resulting tool

The screenshot displays an IDE interface with the following components:

- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard IDE icons for file operations, editing, and running.
- Package Explorer:** Shows the project structure with a JUnit icon and a status bar indicating "Finished after 0.095 seconds", "Runs: 1/1", "Errors: 0", and "Failures: 0".
- Source Editor:** Displays the code for `Lab3Part2Test.java` in the `SD3LabTests` package. The code includes an import for `org.junit.Assert.*` and a `public class Lab3Part2Test` with a `float score = 0;` and a `boolean isRecursive = false;` field.
- Console:** Shows the execution output for `AnimalTest [JUnit] /usr/java/jdk1.7.0_79/bin/java (25 Sep 2015 09:06:17)`. The output includes:
 - Checking if field alive exists... Field alive found
 - Checking if field field exists... Field field found
 - Checking if field location exists... Field location found
 - All required fields are present
 - The alive field is set to true
 - Testing setDead method... The alive field is set to false
 - The setDead method works

[demo during break]

Resulting tool

The screenshot shows an IDE interface with the following components:

- Package Explorer:** Shows the project structure with a test runner for `lab4.foxesAndRabbitsGRAPH.agents.AnimalTest`. It indicates the test finished after 0.015 seconds with 1 failure.
- Code Editor:** Displays the `AnimalTest.java` file. The `test()` method is visible, containing a `try` block that checks for the existence of the `Animal` class and its fields. The `assertTrue` call on line 49 is highlighted, indicating the failure point.
- Failure Trace:** Shows the error message: `java.lang.AssertionError: The Animal class should contain a variable called alive. Remember that variable names are case sensitive.`
- Console:** Shows the output of the test: `<terminated> AnimalTest [JUnit] /usr/java/jdk1.7.0_79/bin/java (25 Sep 2015 09:13:42) Checking if field alive exists...`

[demo during break]



Test **tool** in a
lab followed by **questionnaire**

[can we go beyond **shallow learning?**]



**Extend tool to automate
marking of coursework**

[can we go beyond shallow properties?]