



# Computer Science 507 Software Engineering

The College of Saint Rose  
Spring 2014

## Lab 8: Static Analysis

Due: 6:00 PM, Monday, March 31, 2014

This week's lab exercise will introduce you to a tool that analyzes Java code for potential errors. Of course, it is impossible for a program to find all possible errors in another program. However, there are many common errors that can be found.

You may work alone or in groups of up to size 4 for these exercises.

---

### Static Analysis

For a quick introduction to static analysis and to the tool we will be using this week, read through the slides of a talk by William Pugh at the 2009 JavaOne conference. Don't worry too much about technical details of all examples in the talk – just go for the big picture ideas.

---

### FindBugs Installation and Setup

The tool we will use this week is called FindBugs from the University of Maryland.

FindBugs finds bugs in Java programs, and is itself a Java program. That makes it very easy to install and use either standalone or as an Eclipse plugin. It is up to you how you decide to run FindBugs.

Follow the instructions in the FindBugs Manual to download, install, and run FindBugs on a computer of your choice.

---

### A Simple Example

Copy the Java files from the directory `/home/cs433/examples/javageneric` on `mogul.strose.edu` to the computer where you installed FindBugs. Set up a project, add the directory containing the "javageneric" example code to the project and run an analysis.

#### ? Lab Question 1:

| What errors are reported by FindBugs? (2 points)

#### ? Lab Question 2:

| Are these errors legitimate errors? If not, why not? If so, how could they be fixed? (3 points)

---

### A Larger Example

Next, we will analyze a larger Java example. If you have a Java program that you'd like to use (minimum 10 non-trivial Java classes), please feel free to do so. Otherwise, you can use the "Structure package", the source code of which is available from here. To analyze this one, you'll add the "structure5" directory that is produced when you extract the gzip'ed tar file.

**? Lab Question 3:**

Use FindBugs to analyze your larger example (either structure or another package/program of your choosing). Summarize the results, and choose 3 of the reported problems and describe it in more detail. If it is a legitimate error, suggest a fix. If it is not, explain why it is not. (10 points)

---

**Interaction with JUnit****? Lab Question 4:**

Use FindBugs to analyze your JUnit tests from the previous lab. For each problem reported by FindBugs, describe it briefly. If it is a legitimate error, suggest a fix. If it is not, explain why it is not. (5 points)

---

**Submission**

Before 6:00 PM, Monday, March 31, 2014, submit your lab for grading. Package up all required files into an appropriate archive format (.tar.gz, .zip, and .7z are acceptable) and upload a copy of the using Submission Box at <http://sb.teresco.org> under assignment "Find-Bugs".

---

**Grading**

This lab will be graded out of 20 points.