

CS 474/574 Project

1 Project Description

In this group project you will select a real world open source application, for example, from sourceforge or GitHub, and its features for which you want to define a substantial set of properties. Graduate students can work in groups of 2 people and undergraduates in groups of 3 people. The size of the code in a project (or its submodule) should be between *8K and 12K* lines of code.

Next, you will apply appropriate verification techniques in order to verify the correctness of the program for that set of properties. Your verification approach must consist of at least one dynamic, e.g., testing, and one static, e.g., symbolic execution, technique.

The project, which is 45% of your grade, has the following three deliverables and their grade percentile allocations:

1. Project proposal – 5%
2. Project presentation – 10%
3. Final report – 30%

2 Preliminary Steps

The first step is to form your teams and create private GitHub repositories where each team will be working on their projects. You will add the instructor (esherman77) as a collaborator with Read permission level. Populate the repository with README for now including the names of the team members and potential list of open-source projects. This should be done by **February 2nd**. Using search available in GitHub and sourceforge or using software mining tools such as Boa <http://boa.cs.iastate.edu/examples/index.php> or RepoReapers <https://reporeapers.github.io/results/1.html> to find Java projects of required sizes. If you have some software in mind that satisfied the size requirement then you can also use it.

3 Project Proposal

Your project proposal should be an initial project report document that describes the application that you have chosen to work with. Besides explaining what the program does, it should also include some program's metrics, for example, number of classes, lines of codes. Make sure that you can execute the program on your machine.

In addition, you need to identify what functionality or functionalities of the application you want to verify. Define first in plain English and then using either Java assertion, JML or MOP the set of properties that the program must conform to and/or specify set of behaviors which functionalities must avoid. Provide arguments why those properties reflect desirable and/or undesirable program

behaviors. Do not hesitate to contact the developers if something is unclear about particular functionality. There should be at least non-trivial 20 properties and 5 of them should be sequential properties.

You must submit your project proposals by the end of **February 23rd**. After that, I will review your proposals and meet with each of you to discuss your proposal and take a look at the source code and how you execute the program on your machine.

4 Project Presentation

The dead week is scheduled for project presentation. Each of you will have 15 minutes to present your project. Think about it as a sale pitch, i.e., an advertisement, of your project paper, so that the audience will be interested in reading the full version of your paper. Make sure to include motivation (why it is important), approach (how did you do it) and results (what did you get) in the presentation.

5 Final Report

Your final project report should be as at least six page long conference paper in the official ACM proceedings format <http://www.acm.org/sigs/publications/proceedings-templates>. It is due by the end of **May 6**. Submit the paper only and make sure that the relevant artifacts, i.e., files with Java assertion, JML annotations, and JavaMop files, test suites and parameters and outputs of static techniques can be found in the GitHub repository (mention in README where is what located).

Your paper should be organized as a research paper: introduction, background/related work, approach (defining property, testing and analyzing program to verify those properties and please make sure to argue why you've chosen those approaches) and conclusion.

Share your test cases and findings with the developers of the open source project – almost all project suffers from test case shortages.