The University of Melbourne
Department of Computing and Information Systems
COMP90045 Programming Language Implementation

Assignment 2, 2016

Released: 31 March 2016
PRAZE registration and upload: 12 April at 23:00.
Reviews due: 22 April at 23:00

Objectives

To generate feedback on the parsers and pretty-printers submitted for Stage 1, and to provide an opportunity to learn from other teams' solutions. To practice reading and understanding software. To practice the provision of constructive critique.

Background and context

This is the second stage in a larger task: to write a compiler for a procedural language, Bean. In the first stage, teams submitted parsers and pretty-printers for the language. Each student will now review two randomly allocated project submissions (none of which will be their own). Stage 3 will be to write a semantic analyser and a code generator. The remaining deadlines are

Stage 2a Tuesday 12 April at 23:00. Team effort: Re-submit, anonymised, to PRAZE.

Stage 2b Friday 22 April at 23:00. Individual: Double-blind peer reviewing (via PRAZE).

Stage 2c Friday 29 April at 23:00. Team effort (optional): Feedback to reviewers (via PRAZE).

Stage 3a Friday 6 May at 23:00. Team effort: Submit test data.

Stage 3b Friday 20 May at 23:00. Team effort: Submit compiler.

Peer reviewing using PRAZE

Assignment 1 will be marked as usual and we will provide feedback and a mark. The peer reviewing activity is not a substitute for that and it will not have any impact on marks for Stage 1, only on marks for Stage 2.

This second stage gives you an opportunity to see other solutions and compare them to your own, while preserving some privacy through the anonymity of double-blind reviewing. Thus a central aim is to provide an additional source of learning. Other aims include:

- It should add to (and diversify) the feedback you receive in this subject.
- It should help develop important skills in the reading and understanding of other people's code, as well as in providing and receiving critique in a polite and constructive way.

You can access can review tool, PRAZE, from the COMP90045 LMS site. For the review part, you are *not* asked to assign any marks to the work you are peer reviewing. There is an online review form that will guide you as to how to structure your review and which aspects you may want to assess.

Procedure

Re-submission. PRAZE knows about the students enrolled in COMP90045, and it knows how the class has been split into groups. However, you will need to submit your code again. Only one person per group should submit. The steps required are:

1. Anonymise your code, that is, remove all names, login names, enrolment numbers, etc., from your files—anything that might identify the group.

2. Create a tar archive of all the program files that make up your solution, including a make file called `Makefile`. However, to make sure that the tar file headers *also* leave out identifying details, use the following commands, where `<file1>` and so on are the files you want to package up and submit.

   ```
   gtar cvzf parser.tar --owner=root --group=student <file1> <file2> ...
   ```

   This will create an anonymous tar file and compress it, with the result in `parser.tar.gz`, which is what you will upload to PRAZE. (You later retrieve the compressed file with '`gunzip parser.tar.gz; tar xvf parser.tar`'.)

3. On the COMP90045 LMS site, click 'PRAZE reviewing' on the left-hand menu and follow the link. If asked about a "topic", say "Parser" (PRAZE may or may not ask).

4. Submit the single anonymised file `parser.tar.gz`.

This should be done by the evening of Tuesday 12 April.

Reviewing. On the morning of Thursday 14 April, two submissions will have been allocated for you personally to review. Submit your two reviews to PRAZE by Friday 22 April.

Viewing of related reviews. (Optional.) Shortly after the review deadline you will be able to read the other reviews of the projects that you have reviewed.

Feedback to reviewers. (Optional.) There will be a facility for giving simple feedback to reviewers, rating the usefulness of their reviews. The deadline for this is 29 April.

How to write a good review

An important aim of Stage 2 is to get all teams to be as well prepared for Stage 3 as possible. Since we all work on the same challenge, you should be in a very good position to provide useful feedback to your peers. For example, you will have your own set of test data to use.

Remember to praise positive aspects, and be clear when you highlight weaknesses. A good review not only pinpoints flaws; it also suggests how to repair the flaws. We have set up a review form to guide you through aspects that should be considered when you evaluate code.

Assessment

This peer-review part of the project counts for 6 of the 30 marks allocated to project work. Marks are awarded for the quality of your reviews. For each review task we allocate 0, 1, 2, or 3 marks, according to a scheme that has 0 for no review submitted, 1 for a poor review, 2 for a good review, and 3 for a review that is detailed, careful, and constructive.

Note that a careful review is not necessarily a long review—the amount of feedback will often depend on the quality of the work being reviewed.

Graeme Gange and Harald Søndergaard
30 March 2016