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

Assignment 2, 2020

Released: Tuesday 16 September 2020
Anonymised parser/pretty-printer submission: Saturday 19 September at 23:00
Reviews due: Thursday 1 October at 23:00

Objectives

To generate feedback on the parsers and pretty-printers submitted in 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, Roo. In the first stage, teams submitted parsers and pretty-printers for the language (with a deadline of 18 September). 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 (updated) deadlines are

Stage 2a Saturday 19 September at 23:00. Team effort: Re-submit, anonymised, through Canvas. Review tasks will be available shortly after.

Stage 2b Thursday 1 October at 23:00. Individual: Deadline for completing double-blind peer reviewing.

Stage 3a Monday 19 Oct at 23:00. Team effort: Submit test data.

Stage 3b Wednesday 28 Oct at 23:00. Team effort: Submit compiler.

Code peer reviewing

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 the review tool 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. Since the reviews are intended to be anonymous both ways, you will need to submit your code again. Only one person per group should submit. The steps required are:

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

2. Submit the anonymised files, including the `Makefile`. We are using the "FeedbackFruits" tool through the LMS.

Re-submission should be done by Saturday 19 September at 23:00.

Reviewing. Two submissions will be allocated for you to personally review. You will receive each review task in the form of a zip file. Submit your two online reviews by Thursday 1 October.

Assessment

This peer-review part of the project counts for 4 of the 30 marks allocated to project work. Marks are awarded for the quality of your reviews. For each review task we allocate 0, 0.5, 1, 1.5, or 2 marks, according to this rubric: 0: missing review, 0.5: poor review (minimal effort, fails to recognise clearly strong points, or to identify obvious errors and shortcomings in the submitted code), 1: decent review (shows insight into the reviewed code, but somewhat superficial, possibly misses relevant aspects), 1.5: a good review (making valid points, clearly of value to the reviewees), 2: an excellent review (as for 1.5, plus clear, pedagogical, and to the point).

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 when you assess their code.

Remember to praise positive aspects, and be clear and specific when you highlight weaknesses. A good review is careful and constructive, and if needed it is detailed. But 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. A good review not only pinpoints flaws; it also suggests how to repair them. The online review form will guide you through aspects that should be considered when you evaluate code.

Harald Søndergaard
14 September 2020