

# Compiler Construction 2019

## *Semester project*

### 1 Implementing a pretty printer

For the remainder of the semester, you will be working on an implementation of a Mini Java compiler. In this first part, your task is to familiarise yourself with the template and implement a pretty printer that outputs the source code according to given indentation and formatting rules. The provided JUnit tests are used to verify that your implementation works as expected.

To solve this exercise, do the following tasks.

- Implement a front end for the Mini Java grammar at <http://www.cambridge.org/resources/052182060X/MCIIJ2e/grammar.htm>.
- Implement a pretty printer (<http://en.wikipedia.org/wiki/Prettyprint>) for the given Mini Java grammar that conforms to the provided test suite.
- The provided test suite specifies the pretty printer rules and is mostly for you to get a feel for the number of features you have implemented. It will be used by us to do an initial evaluation of your work, but we reserve the right to use additional test cases and other methods for grading. You are not allowed to modify the test cases in any way.
- You can use the following tools.
  - Eclipse. Mandatory.
  - javacc and its Eclipse plugin, for scanner and parser generation. Mandatory. <http://eclipse-javacc.sourceforge.net>.
  - JTB, for generation of concrete syntax tree and visitors. Optional. <http://www.cs.ucla.edu/~palsberg/jtb/>
- You have time until 29 March 2019, 10:00 to complete this part of the project.
- Push your solution (the Eclipse project, including the parts provided by us) into your Bitbucket repository.
- You are not allowed to use other people's work. All parts of the project solution must be your own work.
- Not all Mini Java features are required for this assignment, but parts of the upcoming project assignments are dependent on the features tested in this assignment, so you are encouraged to do as much as possible.