

Group Work and Collaboration Policies for Compiler Projects

COMPSCI 322 — Computer Languages and Compilers — Spring 2015

February 10, 2015

Group Grading Policy

You will work in a group of 3-4 students to create a solution for each project. I expect each group's members to work together, dividing the work into parts and then integrating the results into a unified, functional product. The final product should result from the combined efforts of all group members, not just one or two "leaders".

All group members will generally receive the same grade for each project. Your group must include a statement in the project report that summarizes how each member contributed to the project. This is meant to ensure that everyone in the group is contributing to the work. In rare cases, I may vary grades for different group members if the contributions were significantly unequal.

Collaboration/External Help Policy

Every member of a group who submits source code or written documents where a large portion of the content is copied from other sources without permission (i.e., not the group members' own work) commits academic misconduct under §UWS 14.03. All members of a group that does so will, at a minimum, receive a grade of **zero** on the assignment with no chance to make up the grade, as allowed by §UWS 14.04(d). This includes multiple groups who turn in substantially identical code and/or reports.

The following rules will help you understand what you are (and are not) allowed to do.

1. Your compiler projects may freely include code from any of the following sources, without citation:
 - (a) Code written by your group members (even if written for other courses)
 - (b) Methods/procedures from your language's standard libraries, including Java API (with Collections Framework), C/C++ standard libraries, C++ STL; treat the Boost libraries for C++ as "standard"
 - (c) Code or pseudocode given in our textbook (*Engineering a Compiler*)
2. You are encouraged to use tools built into your IDE that generate small amounts of code, such as the "generate getters and setters" feature in Eclipse. These can save time and prevent errors.
3. **Get permission from me before using any other external libraries or any tools that generate large amounts of code** (e.g., flex, bison, ANTLR, CUP). I will probably give you permission if you ask, but I need to know in advance how you plan to use these libraries or tools.

Be advised that non-standard libraries and external tools often have non-trivial learning curves. You may find it faster to "roll your own" rather than teach yourself one of these tools — or you might not.
4. You are allowed to use or adapt small pieces of code that you find online in your compiler projects, as you might do for a professional programming job. If you use or adapt code that you found online, you must cite your source by including a comment near the copied/adapted code that contains the URL where you found it. If you are not sure if a piece of code qualifies as "small", please check with me.
5. Groups may discuss general approaches to the projects with the instructor and with other groups, but each group must write its own code and documentation.