Project Team Guidelines and Agreement

Author: Jeffery von Ronne
Course: CS 5363
Semester: Spring 2015

Purpose of Allowing Team Work

In CS5363 students will be completing semester-long compiler projects. These projects may be completed individually or in small teams of 2-4 students. (Two-person teams are recommended.)

The rationale for allowing teams is neither to allow students to take credit for the work of others nor is it to allow students to split the work into semi-independent parts. Rather team work is allowed so that students in a team can sit down together and work together to solve the problems that arise (both with the Scala language and with the algorithms and data structures being used in the compiler). In this way students can learn from each other and obtain a better mastery of the material. The practice of pair programming is highly recommended.¹

¹ I recommend having a look at Williams and Kessler’s article, “All I Really Need to Know About Pair Programming I Learned in Kindergarten.” DOI: 10.1145/332833.332848

Team Work Policies

In order support the pedagogical purpose of group work in this course, certain restrictions apply to students working in teams rather than individually.

No Freeloaders

Everyone on the team must be a regular participant in the work being conducted. All students are responsible for understanding how each part of the compiler works.

If a student’s teammate is not participating fully in the work of the team, students are encouraged to first discuss the issue with the non-participating teammate and the then with the rest of the team. If the teammate is either converted to a contributing member of the team or voluntarily withdraws from the team (notifying the instructor) by the second review deadline, no further action is needed. Otherwise, each team member is individually responsible for notifying the instructor of the non-participation of their teammate.

A student putting their name on work to which they did not contribute will be considered plagiarism. Not notifying the instructor of non-contributing teammates may constitute collusion in the plagiarism of the non-contributing teammate. Falsifying documentation of contributions to cover up a student’s non-contribution is collusion in plagiarism.

Common Coding Sessions

Students working together as a team on the project are expected to spend most of the effort on the project working together on the project at the same computer or on computers physically adjacent to each other.

Towards this end, as part of the Project Team Agreement. The team is asked to identify 5 hours / week that they are all available to meet together to work on the project.

Individual Project Extensions

5% of the course score considered in assigning A grades will be based on an extending the language processed by the core compiler or adding optimizations to the core compiler. There are four options that students can choose from: (1) supporting additional data types, (2) adding recursive procedures, (3) adding SSA-based optimizations, and (4) implementing a Chaitin-Briggs register allocator.
These extensions must be completed individually even if you do your core compiler as part of a team. Furthermore, if you do your core compiler as part of a team, each member of the team must choose a different extension to implement. There is no guarantee that these extensions are equally hard.

To reduce conflicts, you will be asked to choose each team member’s extension as part of the team agreement. You may change these selections later, but only with the agreement of your teammates.

**Documentation**

Each time one or more team members sit down to write code, the work should be recorded in git. That is, each programming session needs to be recorded using one or more git commits. These commits should be on branches that get pushed to the team’s private submission repository. If multiple team members work together on the commit (i.e., when pair programming), they should all be listed in the commit: either as the committer, the commit author, and/or using “Also-by: name <email>” lines at the bottom of the commit message.

**Code Security**

The team members are individually and collectively responsible for ensuring the security of the team’s project code. Each team member should notify the instructor immediately if they suspect or know that another team member has either viewed non-team compiler code while writing parts of the team’s compiler (cheating/plagiarism) or has shared the team’s compiler code with other 5363 students (collusion in cheating).

If the team’s project submissions has code in common with someone else’s project submission, the presumption will be that all member’s of the team were aware of the sharing. An investigation into the cheating, plagiarism, and/or collusion of each team member will proceed on that basis.

**Grade Adjustments**

A grade for the core compiler project submissions will be assigned to the team as a whole. The instructor may make individual (downward) adjustments for students whom the faculty member has reason to believe did not contribute their share to the project effort. This determination may be based on the frequency of authorship in commit messages, reports from other team members, performance on exam questions related to the project or any other information available to the instructor.

**Procedure**

Get together with your prospective team members and discuss how you will work together and keep your code secure. Identify the 5 hours / week when you can work together. Also identify which of the four extension options each team member plans to do.

If you have trouble agreeing on these items, you may wish to consider forming smaller teams — especially if you were planning to form a team of four.

Once you have come to an agreement on these items, complete and sign the agreement form below. To expedite the creation of your group/repository in blackboard and on bitbucket, please scan the agreement form, and email it as an attachment to your TA, Edward.Turpin@utsa.edu. In the body of your email also list the names of the team members.

If you are unable to scan the document, you may also bring it to the course instructor in person, but this may delay the creation of your group/repository.
CS 5363, Spring 2015, Project Team Agreement

We, the undersigned students of 5363, wish to complete the course project as a team. We have read and agree to comply with the policies listed above. Furthermore, we agree that:

1. Each member of the team will participate in the team’s compiler project.

2. We will meet together regularly to work on the project and have set aside at least five hours each week which we are all available for that purpose. These include:

   [List blocks of time totalling at least five hours. Indicate, for each, the day of the week, the starting time, and the ending time.]

3. We understand that the compiler “extensions” will need to be completed individually and no two team members will be allowed to do the same extension. The current plan is for team members to complete extensions as follows:

<table>
<thead>
<tr>
<th>Team Member</th>
<th>Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#1 (additional data types)</td>
</tr>
<tr>
<td></td>
<td>#2 (recursive procedures)</td>
</tr>
<tr>
<td></td>
<td>#3 (SSA optimization pass)</td>
</tr>
<tr>
<td></td>
<td>#4 (register allocation)</td>
</tr>
</tbody>
</table>

   [Deviations from this plan will only be allowed if all team members agree.]

4. We will commit our work into git each after each programming session recording with the commit all of those who worked together in that session.

5. We will maintain the security of our code.

Signed,

<table>
<thead>
<tr>
<th>Printed Name</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>