



## Comp 170 Teamwork Assignments – A Simple Way to Divide the Work

In any team, **\*everyone\* is responsible** for everything that happens and what gets done. It helps to divide some things between different roles and have someone concentrate on certain things. The exact way you work together is up to you!

### 1. Suggested Team Roles

**Requirements** – write a brief statement of what the program will do (unless you have a clear assignment statement provided). Working from these requirements write pseudo code and (if you are doing full OOP) a class diagram. Pseudo code usually goes with each important method (or with main if there is only one method).

**Coding** – work from the requirements to generate code and be sure it compiles. Use proper style. Make sure the code is easy for you (and the whole) team to read and understand. If the pseudo code is not sufficient to understand the code add comments with key parts of the code.

**Test Planning** – before you try to run the program, decide on some specific test cases you will use. Include necessary input and expected output in detail (be precise, e.g. say enter John A. Smith, not enter a name. You make up test cases from the requirements and also from knowing about the kinds of inputs (both correct and incorrect) that the user may enter.

**Code Review** – someone other than the person(s) creating the code should read it carefully looking for both bugs and style / readability problems. They should (depending on the team preference) either make changes themselves or suggest changes back to the original programmer.

**Testing** – carry out the previously defined test cases and report how each once came out. Did the test pass or fail? Were you able to run the test? Were you able to decide if it passes or failed (sometimes the test case will need to be improved to be certain you know what the result should be). Test results should be documented and passed back to the programmer for correction (or to decide it will not be fixed).

Management and Misc. – even a small team has overhead. Who will decide and manage how the team communicates, meets, shares information? How will the team resolve disputes? How will the team enforce discipline, getting things done on time? Successful teams self-manage these topics; often they report to themselves (and others) how they did on meeting obligations.

## 2. Learning about Teamwork

Teams measure themselves in many ways, including feelings. Some things you can reflect on and possibly discuss and document with your team:

- a. Did everyone on the team do about a fair and equal amount of the work?
- b. How would you rate the quality of the program you produced?
- c. About how many hours did you work and how much code did you write? (Teams often measure Source Lines of Code per Person Hour.
- d. How many bugs did you find in testing? Were they all corrected or are some still present in the program?
- e. Given the choice, would each member of the team want to work together again or would they rather work alone?

If the assignment specifies, you may need to record some of these items, keep track of how your team did, and report them as part of the work.

## 3. Grading

Unless the assignment says otherwise, all team members will receive the same grade. Remember the team is responsible for the product. You can divide the authority to work on certain roles but you all share the responsibility for the results.

Comp170 Simple Team Work in Java  
Version 1.0 (original)

©W.L.Honig for Comp 170 Loyola University Chicago, Summer 2017  
//end assignment