### CMPT 415: Project Code Rubric

<table>
<thead>
<tr>
<th>Score</th>
<th>Score</th>
<th>Overall</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1200  | A+ = 95-100 | Exceptional | • Exceptional code contribution  
  ◦ Very many difficult features  
  ◦ >> 2 assignments of code  
  • Consistent development each week  
  • No issues with code quality. |
| 1000  | A = 90-95 | Great | • Great code contribution  
  ◦ Numerous difficult features  
  ◦ > ~2 assignments  
  • Consistent development each week  
  • No issues with code quality. |
| 900   | A- = 85-90 | Good | • Expected amount of code contribution.  
  ◦ A couple important features  
  ◦ ~2 assignments  
  • Consistent development each week  
  • At most a few issues with maintainability or clarity of code. |
| 800   | B+ = 80-85 | Good meets expectations | • Less than expected amount of code contribution.  
  ◦ Some features of significant size successfully implement  
  ◦ ~1.5 assignments  
  • Less consistent development; some weeks with no work.  
  • OK code quality  
  ◦ Code is usable and maintainable  
  ◦ May need some code cleanup |
| 650   | B = 75-80 | Less than expected | • Significant improvement required in terms of amount of code contribution.  
  ◦ Likely only small sized feature or less of work.  
  ◦ ~0.5 assignments  
  • Inconsistent development; may do most work at end of iteration.  
  • Possibly poor code quality.  
  ◦ Demonstrated a low understanding of, or commitment to, code quality. |
| 500   | B- = 70-75 | Significant improvement required | • Extensive improvement required in amount of code.  
  ◦ Virtually no sizable contribution to code.  
  ◦ <~0.5 assignments  
  • Insufficient evidence to warrant a passing grade. Student should talk to instructor to identify how to address deficiencies and earn a passing grade.  
  • Inconsistent development; may do most work at end of iteration.  
  • Possibly poor code quality.  
  ◦ Work below required level for a 3rd year SoSy course. |
Note on Working with Git Code Contribution Score

- It’s similar to Lines of code added (1 point per new line of “real” code).
  - No credit given to changing indentation, etc.
- Give more credit for students who refactor and remove code (vs just changing a line of code).
- Give more credit for hard, complex, or critical code.
- Give more credit for learning new framework, teaching team.
- Adjust mark by work done as role for team (product owner, scrum master, repo manager).
- About \( \frac{1}{2} \) credit given for pair programming.
- Give more credit for design work, helping team-mates, time testing, time doing things not visible directly in code contributions.

Team Roles

- +0 to +10 points given for performance of team role (Scrum Master, Repo Master, Product Owner)
  - Bad: +0
  - Good: +5
  - Great +10
- Usually around +5 in most cases.
  - Repo Manager who spends a lot of time +15.