

# CMPT 373: Project Code Rubric

~LoC	Score	Overall	Description
1200	A+ = 95-100	<b>Exceptional</b>  near best in class	<ul style="list-style-type: none"> <li>• Exceptional code contribution               <ul style="list-style-type: none"> <li>◦ Very many difficult features</li> <li>◦ &gt;&gt; 2 assignments of code</li> </ul> </li> <li>• Consistent development each week</li> <li>• No issues with code quality.</li> </ul>
1000	A = 90-95	<b>Great</b>	<ul style="list-style-type: none"> <li>• Great code contribution               <ul style="list-style-type: none"> <li>◦ Numerous difficult features</li> <li>◦ &gt; ~2 assignments</li> </ul> </li> <li>• Consistent development each week</li> <li>• No issues with code quality.</li> </ul>
900	A- = 85-90		
800	B+ = 80-85	<b>Good</b> meets expectations	<ul style="list-style-type: none"> <li>• Expected amount of code contribution.               <ul style="list-style-type: none"> <li>◦ A couple important features</li> <li>◦ ~2 assignments</li> </ul> </li> <li>• Consistent development each week</li> <li>• At most a few issues with maintainability or clarity of code.</li> </ul>
650	B = 75-80		
500	B- = 70-75		
400	C+ = 65-70	<b>Less than expected</b>	<ul style="list-style-type: none"> <li>• Less than expected amount of code contribution.               <ul style="list-style-type: none"> <li>◦ Some features of significant size successfully implement</li> <li>◦ ~1.5 assignments</li> </ul> </li> <li>• Less consistent development; some weeks with no work.</li> <li>• OK code quality               <ul style="list-style-type: none"> <li>◦ Code is usable and maintainable</li> <li>◦ May need some code cleanup</li> </ul> </li> </ul>
300	C = 60-65		
250	C- = 55-60	<b>Quite a bit less than expected</b>	<ul style="list-style-type: none"> <li>• Quite a bit less than expected amount of code contribution.               <ul style="list-style-type: none"> <li>◦ Likely one medium sized feature or less of work.</li> <li>◦ ~1 assignments</li> </ul> </li> <li>• Inconsistent development; may do most work at end of iteration.</li> <li>• Possibly poor code quality.</li> </ul>
200	D = 50-55	<b>Significant improvement required</b>	<ul style="list-style-type: none"> <li>• Significant improvement required in terms of amount of code contribution.               <ul style="list-style-type: none"> <li>◦ Likely only small sized feature or less of work.</li> <li>◦ ~0.5 assignments</li> </ul> </li> <li>• Inconsistent development; may do most work at end of iteration.</li> <li>• Possibly poor code quality.               <ul style="list-style-type: none"> <li>◦ Demonstrated a low understanding of, or commitment to, code quality.</li> </ul> </li> </ul>
<200	F = < 50	<b>Extensive improvement required</b>	<ul style="list-style-type: none"> <li>• Extensive improvement required in amount of code.               <ul style="list-style-type: none"> <li>◦ Virtually no sizable contribution to code.</li> <li>◦ &lt; ~0.5 assignments</li> </ul> </li> <li>• Insufficient evidence to warrant a passing grade. Student should talk to instructor to identify how to address deficiencies and earn a passing grade.</li> <li>• Inconsistent development; may do most work at end of iteration.</li> <li>• Possibly poor code quality.               <ul style="list-style-type: none"> <li>◦ Work below required level for a 3<sup>rd</sup> year SoSy course.</li> </ul> </li> </ul>

# Marking Adjustments Guidelines

## ~LoC (Lines of Code)

- ~LoC (lines of code) refers to adjusted lines of code added to the project.
- No credit given to code comments, changing indentation, or lines that are just ``.`.
- TA to give more credit for hard, complex, or critical code.
- TA to give about ½ credit given for pair programming.
- TA can adjust for low impact work (easier to write many more tests after some are already created one)

## Frequency of Commits

- Possible ~10% penalty for very infrequent commits / merge requests
- Other penalties possible based on circumstances

## Team Roles (SM, RM, PO)

- Bad: +0%
- Good: +5% (often the value; was helpful)
- Great +10%

## GitLab Issues

- Create/comment on numerous GitLab Issues: +5% to +10%

## Learning

- +5%: a tutorial or two (Ex: learning Bootstrap)
- +10%: significant learning on tutorials, such as learning React for the first time.
- +15%: extensive number of tutorials; learning in multiple technologies / areas.  
For example: learning React, Typescript, CSS, *and* HTML for the first time.

## Other Work

For any of these these below, you must describe your work clearly in your iteration summary. When possible, include proof (link to work, or screen-shot of work). Be clear what you did, vs what you did with others. Range of scores is approximate to show what may be earned based on how extensive the work was.

- UI Mockup / Design: +10% to 30%
- DB Design: +15% to 30%
- REST API design: +15% to 30%
- DevOps (CI/CD, VM , Docker): +10% to 30%
- Video: +5% (writing, narrating, or demoing);  
+20% (creation of a high-quality video)
- Helping teammates:
  - +0: Normal helpfulness
  - +5: Going out of your way to provide help
  - +10 (or more): anchor of the team, going above and beyond to ensure teammates who are really falling behind get the support they need.
- Manual testing: +5% to +15% for significant time spent on testing other people's work

## Code Reviews

- Number of **good quality** code reviews:
  - a couple (1-3) code reviews: +2%
  - some (~4-10) code reviews: +4%
  - many (10-30) code reviews: +8%
  - very many! (31+) code reviews: +10-15%