Code Review

“In the early years of programming, a program was regarded as the private property of the programmer. One would no more think of reading a colleague’s program unbidden than of picking up a love letter and reading it. This is essentially what a program was, a love letter from the programmer to the hardware, full of the intimate details known only to partners in an affair. Consequently, programs became larded with the pet names and verbal shorthand so popular with lovers who live in the blissful abstraction that assumes that theirs is the only existence in the universe. Such programs are unintelligible to those outside the partnership.”

—Michael Marcotty

“Code review is like a pair of pants. If you work at home, pants are optional. But if you are in public, you’d better make sure you have pants on, even if no one looks at you.”

—Sander Sade

Assigned on Tuesday, Feb. 7  
Marked by TA = 2/2
Due on Monday Feb 13  
Total marks = 2/100

1 Objective

Most software is developed in large projects with many developers, and needs to be maintained for years. Consequently, writing readable and maintainable code is an essential skill. In this assignment you will write a review of code created by others in order to:

1 | See what makes it easy or hard to understand code you didn’t write.

2 | Learn how to give feedback to improve a programmer’s style.

2 Detailed Instructions

In milestone 1 you and your teammates wrote code to implement an API for map data. In this assignment, you and your teammates will review another team’s code submission for milestone 1.

You should read the code provided to you, and write a brief document giving feedback on what you see as good and bad style and design within the code. The code for you to review will be attached to your wiki page by your TA. You will write one document as a team, and upload it to your wiki page for your TA to mark.
The code review document you write must be a maximum of one page long. If you wish to include longer examples of code highlighting what you see as specific instances of good or bad style, you can also attach an appendix with fragments from the code you were given.

3 Helpful Hints

- Read the code you are given to understand its overall structure, not every detail. You are only assessing if it is well-structured and readable, not debugging it.
- Discuss the code with your teammates once they have also read it.
- Refer to the lecture notes on good coding style for ideas on good and bad style, but feel free to also give other feedback on the code.
- Be succinct, and follow the Made to Stick rule: Simple
- Give specific examples from the code of good or bad style; in Made to Stick terms be Concrete.
- Your most important feedback should come first, to make sure the reader sees it; in Made to Stick terms, don’t bury the lead.

The lecture notes have many suggestions for writing good and readable code that are relevant here.

4 Grading

Your TA will grade the quality of your code review. Your grade will reflect both how well you assessed the code and how clearly you write to convey your feedback. Ideally you will point out multiple ways in which the code displays good style and readability and/or where the style could be improved, with specific examples of each.

Your TA will ask some follow-up questions to individuals in the lab period after the review is marked to ensure all team members participated in the code review and can explain their feedback. Your answers to these questions will also impact your mark.