ECOR1051 Peer Code Review Report

Each individual within a team shall review the code written by another person in the team. Conversely, every person shall receive feedback about their code from a teammate. That feedback shall be presented in this report. Edit the document, save it and submit it.

Required Filename: xxx-P6-aaa-bbb.docx (or .pdf)

xxx is your group identifier

aaa is your surname (not your first name)

bbb is the surname of the person whose code you are reviewing (not their first name)

# Identifying Information

The reviewer must complete the following fields:

Team Identifier:

Full Name of Reviewer :

Full Name of Person who wrote the code that is being reviewed:

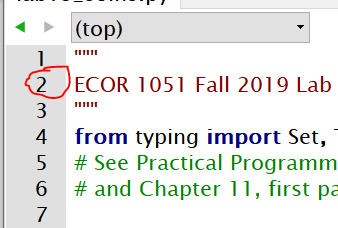
Name of Files that are being reviewed:

# Tabulation of Errors

The table on the next page is to be completed while studying the code written by your teammate. Your review must be guided by the criteria provided in the posted **Code Review Checklist**. Following is an explanation of the columns to be completed in the table:

**Filename**: In the case of multiple files to be reviewed

**Line Number**: Indicate which line of code you are commenting on.

* Open the code within Wing IDE and use the line numbers on the side.
* 

**CheckList Item**: This field must match up with the item identifiers within the posted **CodeReview-Checklist.**

Example: **CL1** refers to *Indentation – Continuation lines should align vertically or use a hanging indent.*

Example: NC1 refers to *Variable names are meaningfully descriptive*

**Description / Note / Suggested Fix**: Minimally, you must clarify the error you are identifying. Hopefully, you will also suggest how to fix it.

In this table, you may have multiple entries – at different line numbers – of the same error. That is okay.

|  |  |  |  |
| --- | --- | --- | --- |
| **File name** | **Line #** | **Checklist Item** | **Description / Note / Suggested Fix** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | Add more rows as needed |

**Summary of Errors**

Write a 3-5 sentence summary of the errors that you found in your teammate’s code.

* Perhaps many of the errors were examples of the same kind (e.g. Always hard-coding values?)
* Or perhaps the errors signal an overall weak approach (e.g. brittle or smelly code).
* Look at your table of errors and group them together.

If your teammate’s code is so good that you did not find enough errors to comment upon, write a 3-5 sentence summary of why you think their code is exceptional.

**Overall Evaluation (Unsatisfactory, Marginal, Satisfactory)**

We’ve been using this 3-value rating system everywhere in this course, and you should have an intuitive feel for it by now. Demonstrate your understanding by grading your teammate’s code (U, M, S). You must justify your evaluation with a reason or two.