

GROUP INSPECTION

Recorder: _____

Reader: _____

Other reviewer(s): _____

Code author: _____

Instructions: The Author should briefly present his or her code and point out any noteworthy elements or known problems. The Reader should then read the code aloud, line-by-line. At any point, any Reviewer (including the Reader and Recorder) may offer comments based on their individual inspections. The Recorder should make notes of such comments here:

GENERAL NOTES & COMMENTS:

Once the code has been reviewed, the Reviewers should discuss their findings and come to a consensus on the following questions. The Recorder will record the group's conclusions.

SYNTAX

1. Compiles. Does the code compile?

YES – Compiles

NO – Few/minor errors

NO – Many/major errors

Comments:

SEMANTICS

2. Correctness. Is the program complete? Is the program output correct? For instance, when the program performs a calculation, does it produce the right answer? If the program produces incorrect output, please describe which cases or what input leads to the error.

YES – Complete; no errors

NO – Complete; single error found

NO – Incomplete and/or multiple errors found

Comments:

3. Runtime errors. Does the program crash at any time? If so, please describe the input or situation in which the crash occurred. Consider unusual or border cases, such as negative numbers, zero, empty strings, invalid menu choices, etc.

NO – No errors

YES – Single error found

YES – Multiple errors found

Comments:

4. Elegance. Did you notice any particularly good error handling? Were there any particularly clear or elegant sections of code? In contrast, was there a lot of superfluous code—large sections of repeated code, unused variables, inefficient code, etc.

GOOD – Elegant and robust

OK – Nothing noteworthy either way

POOR – Unnecessary or inefficient code

Comments:

PRAGMATICS

For the following statements, note your agreement according to the following scale:

5 – Strongly Agree 4 – Agree 3 – Neutral 2 – Disagree 1 – Strongly Disagree

Code Formatting:

5. The code was easy to read. 5 – 4 – 3 – 2 – 1
6. The comments and documentation were clear and helpful. 5 – 4 – 3 – 2 – 1

Comments:

User Interface:

7. The purpose/function of the program was clear. 5 – 4 – 3 – 2 – 1
8. Instructions to the user were clear. 5 – 4 – 3 – 2 – 1
9. It was easy to enter input correctly. 5 – 4 – 3 – 2 – 1
10. Error messages were clear, helpful, and explained what was wrong. 5 – 4 – 3 – 2 – 1
11. The program was generally easy to use. 5 – 4 – 3 – 2 – 1

Comments:

Features:

12. Were there any extra features beyond what was required for the assignment? If so, what were they?

- 12b. The extra features were valuable additions to the program. 5 – 4 – 3 – 2 – 1

Comments: