

INDIVIDUAL INSPECTION

Reviewer (your name): _____

Code author: _____

Instructions: Compile and run the code (if possible). Then examine the details of the code and try running the program with different inputs to check the different cases handled by the code. Based on your examination, answer the following questions by circling the best answer. If you detect any errors, potential problems, or other noteworthy features, describe them in your comments below.

SYNTAX

1. Compiles. Does the code compile?

YES – Compiles

If not, is this due to only a couple syntax errors you can easily correct? Or does the program suffer severe syntax problems?

NO – Few/minor errors

Comments:

NO – Many/major errors

SEMANTICS

2. Correctness. Is the program complete? Does the program perform as expected? Is the program output correct? 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

Comments:

NO – Incomplete and/or multiple errors found

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. Variables names were descriptive and relevant. | 5 – 4 – 3 – 2 – 1 |
| 7. Code was properly indented so that it was easy to match up braces. | 5 – 4 – 3 – 2 – 1 |
| 8. 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 – NA |
| 9. It was easy to enter input correctly. | 5 – 4 – 3 – 2 – 1 – NA |
| 10. Error messages were clear, helpful, and explained what was wrong. | 5 – 4 – 3 – 2 – 1 – NA |
| 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 – NA |
|---|------------------------|

Comments: