

General Usage Rubric – CS 249

(Update: 08 July 2022)

Most grading will be conducted by applying the rubric specified for the project.

However, for those issues outside the specific project rubric, the rubric below will be applied.

1. use of non-self-documenting or single-letter variable: -1 per declaration and every use
2. missing or non-aligned curly braces: -1 per pair/occasion
3. redundant boolean test: -1 per occasion
 - a. e.g., `if(<boolean expression> == false)`
 - b. e.g., `if(<boolean expression> == true)`
4. second, or subsequent, `if` statement that should logically be `else`: -2 per occasion
5. any data/state change in array brackets or in function parameters: -2 per occasion
 - a. e.g., `value = array[index++];`
 - b. e.g., `value = someFunction(otherValue, myValue--);`
 - c. note: `array[index + 1]` or `someFunction(otherValue, myValue - 1)` are appropriate and acceptable
6. declaration of variable within any loop: -2 per occasion
 - a. e.g., `for(int index = 0; ...)`
 - b. e.g., `while(someCondition)`

```
{
    int newValue = someValue;
    ...
}
```
7. use of incomplete `for` loop (e.g., `for(; index < 5; index++)`): -2 per occasion
8. use of `if/else` in place of single Boolean `return` statement: -2 per occasion
9. any code on the same line as a curly brace: -2 per occasion
10. use of 1 or 0 (or any other numbers) in place of `true` or `false`: -2 per occasion
11. use of literals (e.g., `ints`, `chars`, etc.) when constants or variables should be used: -2 per occasion
 - a. Note that in most cases, string literals are more readable than constants or variables so string literals are recommended unless the same string value is used in several places
 - b. Judgement: If it would be redundant or unreasonable to create a variable or constant for a given literal, just use the literal
12. use of `break` anywhere but in a `switch` operation: -2 per occasion
13. use of global variables anywhere in the program: -3 per occasion
14. use of tabs causing readability issues: -3 (one time)
15. use of `return;` (without value): -3
16. use of `continue` anywhere: -3 per occasion
17. any I/O in a function not specified for I/O operations: -3 per occasion
18. use of functions in parameter lists or array brackets: -3 per occasion
19. creation or use of empty `if` or `else` blocks: -3 per occasion
20. code more than 80 characters: -5 (one time)
21. use of disallowed programming operations (-5 per occasion)
 - a. Includes operations that have not been discussed in class at the point of usage
 - b. Includes operations that have not been found in the online reference at the point of usage
 - c. other examples: ternary operator, `goto`, `for/each` (enhanced) loop, etc.
22. use of unspecified utility functions and/or data: -5 per occasion
23. --- new items will be added as needed ---