Worksheet 7:

1. A) No, because a global constant cannot be modified, and is just used as the same value wherever it is used. It does not create a modularity problem.

B) By requiring more parameters, a method/function’s coupling increases/tightens. This is because the calling method requires more specific parameters that may be specific only to one single call. A method with low coupling should be able to be reused for different purposes.

C) We want to reduce the total line count of a program, to improve cohesion within the program. By removing code duplication and instead writing methods for said code, we can create code that is easier to read, maintain, and test. It may not be possible to have code with no duplication at all, due a specific function needing a slightly modified version of some code.

1. Checklist:

* 1. Does each module carry out a singular task?
* 2. Does every module have under 6 imports?
* 3. Are there no global variables used?
* 4. Is there a lack of control flags?

1. Review

* Three global variables are used. (3)
* Minmax module carries out two tasks (1)

1. Refactoring

* Global variables are defined in main and passed into methods where necessary, however I believe they can be removed.
* Minmax will be split into two separate methods. Min() and Max().
* Each method using the global variables will be rewritten to include return statements.

Cohesion: The extent to which a single module does one well-defined task. Want it to be high

Coupling: High coupling means that the contents of each method rely heavily on each other. Want it to be as loose as possible.

1. Continuing change: a useful program either undergoes continual change/evolution, or becomes progressively less useful.
2. Increasing complexity: as a program changes, its design complexity increases and its structure degenerates, unless extra work (refactoring) is done to compensate.