Carlos Sathler | cssathler@gmail.com
INFO-I 535 – MGMT ACCESS USE BIG DATA (Fall 2016)

Exercise: 1. - Ivory writes a program with 20 modules of code. Each module contains several procedures. In the first implementation of her program, she finds that each module contains at least one call to every other module. Each module contains 100 lines of code.

1. How long is Ivory's program (in lines of code)?
   N = lines of code (LOC) per module = 100
   K = number of modules = 20

   Total LOC = 100 x 20 = 2,000

2. How many module interconnections are there in the implementation? Note that each call from one module to another is an interconnection.
   Each of the K modules "calls" K-1 modules; K = 20

   Total interconnections = K x (K-1) = 20 x 19 = 380

Exercise: 2. - Ivory decides to reorganize her program into 4 main modules, each containing four submodules in a one-level hierarchy. The four main modules each have calls to all the other main modules, and within each main module, the four submodules each have calls to one another. There are still 100 lines of code per submodule, but each main module needs 100 lines of management code.

1. How long is Ivory's program now?
   Each main module = 100 LOC
   4 sub-modules under main module = 4 x 100 = 400 LOC
   4 main modules with 4 sub-modules each = 500 x 4 = 2,000

   Total LOC = 500 x 4 = 2,000

2. How many interconnections are there now? Include module-to-module and sub module-to-sub module interconnections.
   Main module connections = 4 x 3 = 12
   Sub-module connections = 4 x (4 x 3) = 48
   Total connections = 12 + 48 = 60

   Note: while I think that is the answer the problem is seeking, it is a simplification; it leaves out the fact that each main module needs to invoke at least one of its sub-modules. So there will be at least one additional connection between each main module and at least one of its sub-modules. With that in mind, an alternative reasonable answer would be 64.

3. Was using hierarchy a good decision? Why or why not?
   It was a good decision because there is a significant reduction in the number of interfaces between modules. It will be much easier to design, build and test each module of the system, since each interacts with fewer modules.