ECE 421 Assignment 3

1. An object is something that can be compared to another object of the same type, giving a less than, equal to, or greater than result.

2. Objects should be able to be supplied through the command line, or via a text file.

3. The sort is generic, in that it accepts a comparison function as a parameter. The comparison is user definable, and must return a number as the result of the comparison. It should inherit from Object to be reusable

4. On a uniprocessor system, a multithreaded implementation will likely run in a sequential fashion, executing threads in a round robin style. This will result in the system scaling the same as a non-threaded approach, with the thread generator acting as the top of a recursive function call stack. Since we are implementing merge sort, it will grow approximately according to t=nlog(n), t being time and n being number of threads.

5. The Exceptions Tidy Threads pattern is applicable, as it deals with cleanup after thread exceptions. Module Errno maps C/OS style error codes into an exception subclass that is compatible with ruby. This will be useful for things like files or directories not existing. RuntimeError, ThreadError, and StandardError are applicable to our problem. Upon file errors, the system will exit and inform the user without performing any operations. Thread errors will cause the system to fail, which will halt the program. It will not begin to produce output until the objects have been successfully sorted, so there will be nothing to clean up.

6. Thread based solutions require only one memory space and set of variables, which all threads can access and update with locks where appropriate. Processes require global constructs to access data between each other, as they all have their own copy of the memory space at the time of creation. This has caused our solution to utilize small worker threads being created from a master thread which contains all the necessary information.

7. The recursive calls to mergesort must be synchronized with the calls to merge, as the merging can’t begin before the recursive calls are finished. As well, merge itself has recursive calls to merge two sublists, which must be synchronized with other calls to merge. An exception in a thread will cause a system failure, which will be caught by the thread generator. This will notify the main thread which will in turn notify the user. The remaining threads will then be stopped and no output file will be created.

8. The contracts written for parts 1 and 2 represent a sort of configuration management, specifying the behaviour of the system. We have been using git and github for version control for the previous assignments.

9. a) Refactoring is restructuring code into more logical configurations, while maintaining its functionality.

b) No, once our software has been written, we hand it in. It's already 5:00.

c) n/a

d) We could have used the parameterized method pattern in assignment 2 for the file watchers, making just one function that accepted the type of watching to be done as an argument. We also could have replaced the constructor for the SparseMatrix with a polymorphic constructor instead of using conditionals.