C201
Turning in Program Assignments

1) When turning in programming assignments, the following items should be included:

   a) A listing of the source code on continuous form paper or on **stapled** sheets. Please make sure you use staples, no paper clips.

   b) A printout that records enough test runs of the program to convince me that your program logic is correct. The basic idea of testing is to try to prove the program works in all possible cases. It is normally impossible to test for all cases, so you must develop a good testing strategy and carefully choose data to be used. Do not wait until the last minute and just try numbers! **Using a large number of test runs with no clear strategy will result in lost points. Also not enough test runs will lose points.**

   c) The printout **MUST** contain Handwritten Annotations that help me to understand your testing strategy. Say why you chose each test datum. If no annotations then fewer points will be given. Put the annotations right on the printout itself, and even draw lines to separate different test data areas. **Make output easy to read and follow.** The program and output are like a report and hence should be well organized and easy to follow and look nice as well.

2) Program testing should normally include:

Normal Case:
Test data that is most likely to occur when program runs. Test this data in some logical order, and each item tested should be included for a reason. Remember when testing characters, characters are characters, i.e. "123" vs "abc" are treated the same.

Special Case:
Using the knowledge that you have as author of the algorithm, test for data that might "break" your algorithm. For example, test for off-by-one errors by choosing numbers that are close to those that play a crucial role in your algorithm. Other numbers may be endpoints for test data, i.e. integers between 0 and 100, make sure you test 0 and 100. If program divides by 30 then test 30.

Illegal Case:
Some assignments will require your program to deal with illegal or empty data sets. Data may be illegal because it is formatted incorrectly or because it is out of range. If your program is written to handle illegal data, it should deal with this data gracefully. If your program was not written to handle illegal data, then this should be noted in the program comments and such illegal data should not be tested.