Notes:

- Read Course Information: Section 6 (Miscellaneous) and Section 8 (Academic Dishonesty or Misconduct).
- When you are giving a construction, example, etc., provide a justification with your argument. Your solutions to numerical problems must contain the derivation of your answers. In all of your presentations, strive for correctness, completeness, and clarity. When in doubt about the assumptions of problems, the interpretations of wording, etc., consult the instructor.
- You should strive to complete all problems assigned, and a subset of them will be graded.

1. Read the notes above carefully.
2. Do [Cha10] Chapter 1, Section 1.4, exercises 1, 4, and 5(a).
3. Do [Cha10] Chapter 2, Section 2.1, exercises 1 and 4.
5. In [Cha10] Chapter 2, section 2.2: Machine Epsilon, it is stated that “The maximum relative error made in representing any mathematical real number in a given floating point number system is, assuming overflow and underflow do not occur, no greater in magnitude than machine epsilon.”
   Is the statement correct? Justify your answer.
6. [Programming Problem] Due Date: February 25 (Monday), 2019
   Follow the specification/requirement given in the problem, for example, follow the program segments in Section 2.6 for computing rightGap, leftGap, flBase, and numDigits.
   Your main program encapsulating the implementations of the necessary packages/subprograms and testing code should incorporate adequate documentation and good programming styles. (When in doubt, please ask.) Also, subprograms should be properly documented with objectives, pre- and post-conditions. Also, read [Cha10] Prependix B: Programming guidelines.
   Develop a simple plan for implementing, testing, and debugging your program. Use bottom-up testing to complement the top-down approach.

Hand-in

(a) Submit the discussion part to the instructor.
(b) Submit your source program(s) [with header: CS 3513: Program 1] electronically to our teaching assistant (reza.marzban@okstate.edu) on our departmental machine (csx.cs.okstate.edu). Since our grader will compile and test run your programs on our departmental machine, you should compile your source programs and test run them on it before your submission.