Read all of Chapters 3 and 4.

0.1 Read and solve, but do not turn in: Book, 2.16. [CFLs closed under $\cup$, $\circ$, $\ast$]
   Solve by using both CFLs and PDAs.

0.2 Read and solve, but do not turn in: Book, 2.18. [CFL $\cap$ regular = CFL]
   You can check your solution with the one in the book.

0.3 Read and solve, but do not turn in: Book, 2.26. [Chomsky normal form]
   1. Let $\Sigma = \{0, 1, \#\}$. Let $C = \{x\#x^R\#x \mid x \in \{0, 1\}^*\}$.
      (a) Prove that $C$ is not a CFL.
      (b) Let $D = C \cup \Sigma \Sigma (\Sigma \Sigma \Sigma)^*$. Show that $D$ is a CFL.
      (c) Let $E = C \cup \Sigma (\Sigma \Sigma \Sigma)^*$. Show that $E$ is not a CFL.

2. For this problem you may assume the result of Problem 1. Let $\overline{C}$ be the complement of $C$ from that problem. Again $\Sigma = \{0, 1, \#\}$.
   (a) Show that $\overline{C}$ is a CFL by giving either a CFG or a PDA. If you give a CFG, then include comments that explains how your grammar works.
   (b) Show that the class of CFLs is not closed under complement.
   (c) Show that the class of CFLs is not closed under intersection.

3. Let a $k$-PDA be a pushdown automaton that has $k$ stacks. Thus a 0-PDA is an NFA and a 1-PDA is a conventional PDA. You already know that 1-PDAs are more powerful (recognize a larger class of languages) than 0-PDAs.
   (a) Show that 2-PDAs are more powerful than 1-PDAs.
   (b) Show that 3-PDAs are not more powerful than 2-PDAs.
      (Hint: Simulate a Turing machine tape by using two stacks.)

4. A useless state in a pushdown automaton is never entered on any input string. Consider the problem of determining whether a pushdown automaton has any useless states. Formulate this problem as a language and show that it is decidable. (Hint: Use a theorem from lecture to give a short solution.)

5. Show that a language is decidable iff some enumerator enumerates the language in string order. (String order is the standard length-increasing, lexicographic order, see text p 14).

6. Let $C$ be a language. Prove that $C$ is Turing-recognizable iff a decidable language $D$ exists such that $C = \{x \mid \exists y \in \{0, 1\}^* ((x, y) \in D)\}$. (Hint: You must prove both directions of the “iff”. The $(\leftarrow)$ direction is easier. For the $(\rightarrow)$ direction, think of $y$ as providing additional information that allows you to confirm when $x \in C$, but without the possibility of looping.)

7.* (optional) Show that every infinite T-recognizable language has an infinite decidable subset.