No books, no calculators. Please answer **four** of the following numbered questions, on a separate paper. Please use a maximum of one page per answer.

1. Let $\Sigma = \{a, b, c\}$. For the following languages over Σ, draw a DFA or, if not possible, a PDA, and if this is not possible either, then a Turing machine. Also: Explain, by use of the pumping lemma for regular languages, why the language in question is not regular, if you think it is not.
   (a) $\{a^n b^n c^k \mid n, k \geq 0\}$
   (b) $\{ww \mid w \in \Sigma^*\}$
   (c) $\{a^n b^m c^k \mid n \equiv m \mod 3 \wedge m \equiv k \mod 2\}$
2. Pick an **NP**-complete problem, describe it and prove that it is, in fact **NP**-complete. Unless your problem is a variant of SAT, you can use reduction from SAT.
3. Consider the following complexity classes: **P**, **NP**, **PSPACE**, **NL**, **L**. Give a short definition for each of them, and indicate how they relate to each other. Give two examples of problems that belong to two classes. Mention two theorems relevant to either the relationships between the classes or characterizations of the classes.
4. Pick a problem you know to be undecidable and give its definition. Explain why the problem is undecidable either by a direct proof or by reduction from a well-established undecidable problem. Then mention at least one other undecidable problem as another example.
5. Consider the following operations. Indicate, whether the result is a context-free language, by giving a construction of PDA or a CFG, or by using the pumping lemma for CFLs to show it is not.
   (a) Union of two context-free languages A and B.
   (b) The complement of a context-free language A. (I.e., words that are not A)
   (c) The *reversal* of a context-free language A, i.e., words that are in A, but backwards.
6. Explain the Church-Turing thesis. Give some examples of models of computation and their relation to Church-Turing thesis. Can the thesis be proven? Why, and why not?