1. Functional vs. Imperative vs. Logic Languages:
   What do you think are the main distinguishing characteristics between functional, imperative, and logic languages (one per language class is sufficient)?

2. Rules in Prolog:
   Restate the following Prolog rule using (universal and existential) quantifiers.

   \[
sibling(X, Y) :- \text{mother}(M, X), \text{mother}(M, Y), \text{father}(F, X), \text{father}(F, Y).
   \]

3. List manipulation:
   Scheme uses \texttt{car} and \texttt{cdr} for extraction of the head and tail of a list. What is the equivalent in Prolog?

4. A simple database in Prolog:
   Consider the following set of facts and rules in the Prolog database:

   
   \[
   \begin{align*}
   \text{takes}(\text{jane\_doe, his201}). \\
   \text{takes}(\text{jane\_doe, cs254}). \\
   \text{takes}(\text{ajit\_chandra, art302}). \\
   \text{takes}(\text{ajit\_chandra, cs254}). \\
   \text{classmates}(X, Y) :- \text{takes}(X, Z), \text{takes}(Y, Z).
   \end{align*}
   \]

   The query

   \[
   ?- \text{classmates}(\text{jane\_doe, X})
   \]

   will succeed three times: twice with \(X = \text{jane\_doe}\) and once with \(X = \text{ajit\_chandra}\). Show how to modify the \text{classmates}(X, Y) rule so that a student is not considered a classmate of him or herself.