Recall the (informal) definition of logical entailment:
A collection of sentences $S_1, S_2, ..., S_n$ logically entails another sentence $S$ if the truth of $S$ is implicit in the truth of the $S_i$ sentences. (Therefore, and closer to the formal definition, the meaning of the terms in the $S_i$ sentences do not matter in determining whether $S$ is logically entailed by $S_1, S_2, ..., S_n$.)

Consider the following knowledge base (KB), written as a Prolog program.

\begin{verbatim}
dog(X) :- poodle(X).
dog(X) :- collie(X).
poodle(X) :- poodle(X).
collie(fido).
\end{verbatim}

Recall that Prolog uses back-chaining in answering queries.

1. The KB above logically entails \texttt{dog(fido)}. True or false? \textbf{Answer}: True.
2. The query \texttt{dog(fido)} will not succeed. True or false? \textbf{Answer}: True.
3. Back-chaining is complete on Prolog KBs. True or false? \textbf{Answer}: False.