assert(X) and retract(X)

assert(X)  Adds a new fact or clause to the database. Term is asserted as the last fact or clause with the same key predicate.

asserta(X) Same as assert, but adds a clause at the beginning of the database
assertz(X) Exactly the same as assert(X)

‘a’ being the first letter and ‘z’ being the last letter of the alphabet should remind you where in the database you are adding a new fact or a clause.

retract(X) removes fact or clause X from the database.

retractall(X) removes all facts or clauses from the database for which the head unifies with X.

For example:

assert(good(skywalker, luke)).
assert(good(solo, han)).
assert(bad(vader, darth)).

?- listing(good).
:- dynamic good/2.
good(skywalker, luke).
good(solo, han).
Yes.

?- retract(bad(vader, darth)).
Yes

?- listing(bad).
:- dynamic bad/2.
Yes

?- retractall(good(_, _)).
Yes

?- good(X, Y).
No
Something you might run into (although the project was specified in such a way that you should not):

**No permission to modify static_program**

You can use assert to add new facts at any point within the program, but the interpreter will complain (ERROR: Undefined procedure: x/1) if you try to redefine an existing definition after the program is loaded. You can use the predicate dynamic/1 to enable redefinitions. For instance,

```prolog
?- [likes].
% likes compiled 0.00 sec, 2,220 bytes

Yes
?- assert(american(burger)).
Yes
?- indian(X).
X = curry ;
X = tandoori ;
No
?- assert(indian(bengali)).
ERROR: No permission to modify static_procedure `indian/1'

?- dynamic(indian/1).
Yes
?- assert(indian(bengali)).
Yes
?- indian(X).
X = curry ;
X = tandoori ;
X = bengali ;
No
?- 
```

[explanation by Félix Hernández-Campos, UNC Chapel Hill]