

```
declare(Var,Val) :-
    retract(binding(Var,_)),
    assert(binding(Var, Val)).
declare(Var,Val) :- assert(binding(Var,Val)).

bind(Variable,Value) :-
    retract(binding(Variable,_)),
    assert(binding(Variable, Value)).

valueOf(Variable,Value) :- binding(Variable,Value).

undeclear(Var) :- retract(binding(Var,_)).

inc(Variable) :-
    retract(binding(Variable,Value)),
    NewValue is Value + 1,
    assert(binding(Variable, NewValue)).

dec(Variable) :-
    retract(binding(Variable,Value)),
    NewValue is Value - 1,
    assert(binding(Variable, NewValue)).

add(Variable,Number) :-
    retract(binding(Variable, Value)),
    NewValue is Value + Number,
    assert(binding(Variable, NewValue)).

displayBindings :-
    binding(Variable,Value),
    write(Variable),
    write(' => '),
    write(Value), nl, fail.
displayBindings.

prepend(Variable,Value) :- %assume a list!
    retract(binding(Variable,OldValue)),
    NewValue = [Value|OldValue],
    assert(binding(Variable,NewValue)).
```