

```
%%% FILE: gvl.pro
%%% TYPE: Prolog source
%%% Line: very simple global variable ADT
%%% Date: 9/23/17
```

```
%% Essential functionality
```

```
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).
```

```
undeclare(Var) :-
    retract(binding(Var, _)).
```

```
%%binding display functionality
```

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

```
%%Arithmetic operator functionality
```

```
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)).
```

```
sub(Variable, Number) :-
    retract(binding(Variable, Value)),
    NewValue is Value - Number,
    assert(binding(Variable, NewValue)).
```

```
mul(Variable, Number) :-
    retract(binding(Variable, Value)),
```

```
NewValue is Value * Number,  
assert(binding(Variable, NewValue)).
```

```
div(Variable, Number) :-  
  retract(binding(Variable, Value)),  
  NewValue is Value / Number,  
  assert(binding(Variable, NewValue)).
```