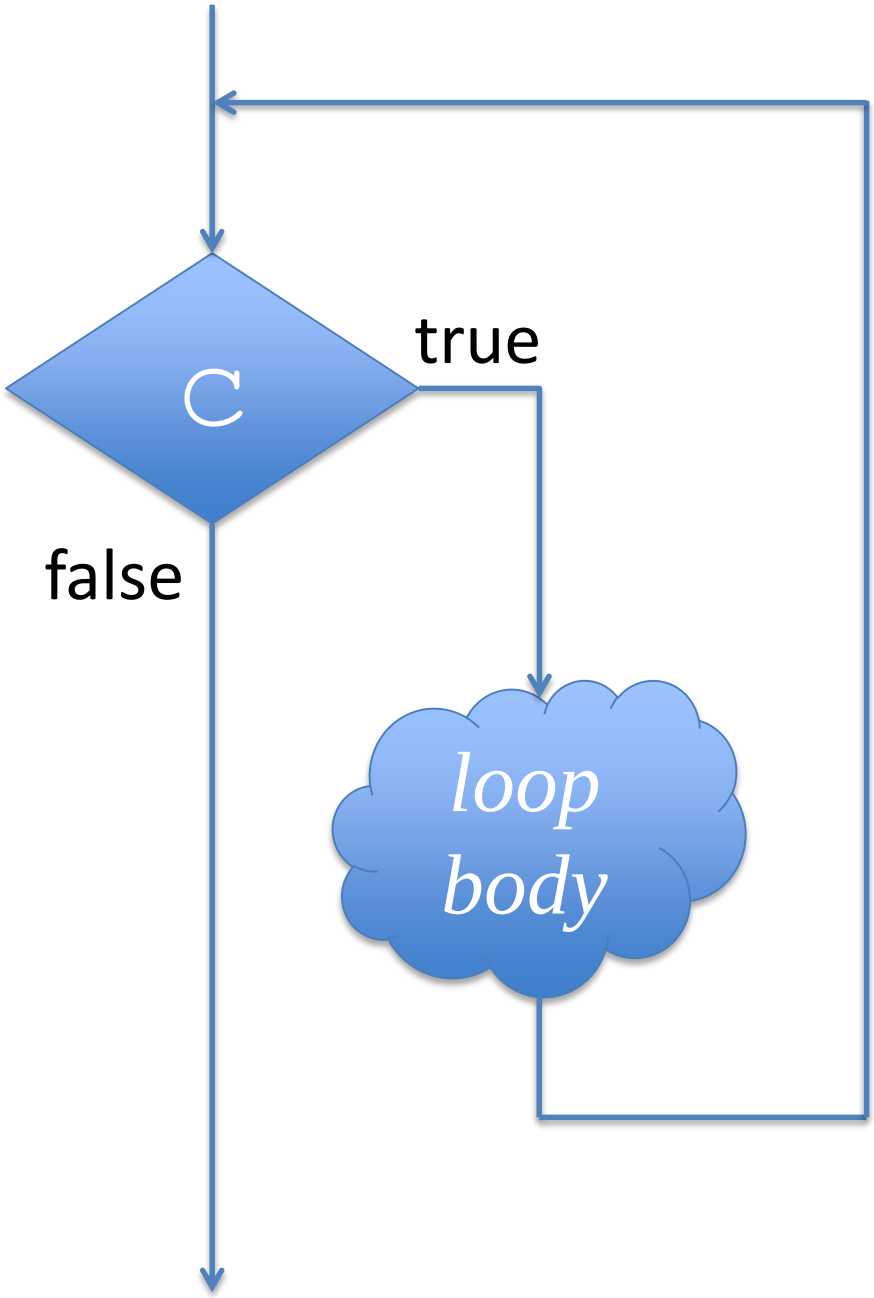
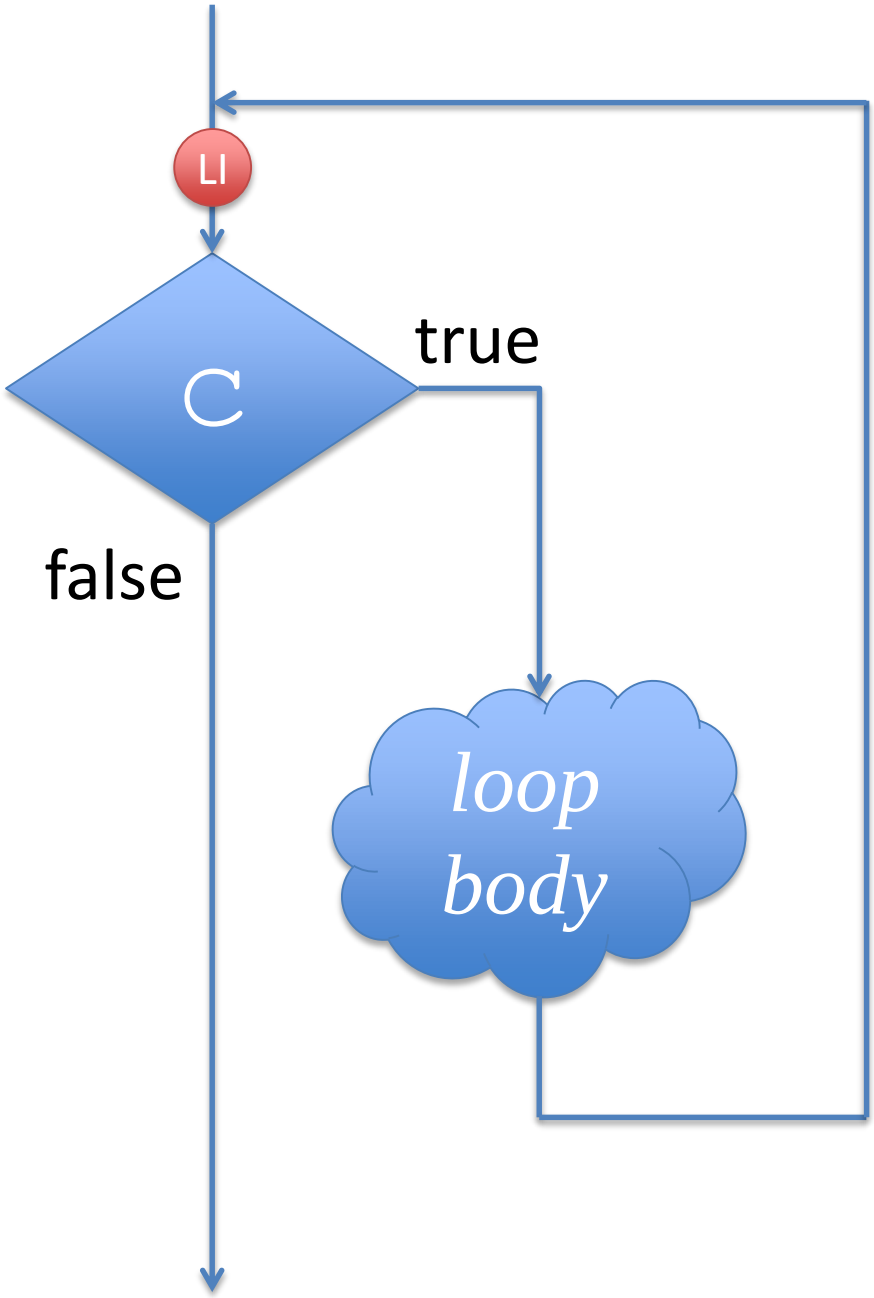



```
while (C) {
    loop body
}
```

Loop Invariant

A boolean condition that is checked *immediately before every evaluation of the loop guard*.




```
while (C)
//@loop_invariant LI;
{
    loop body
}
```

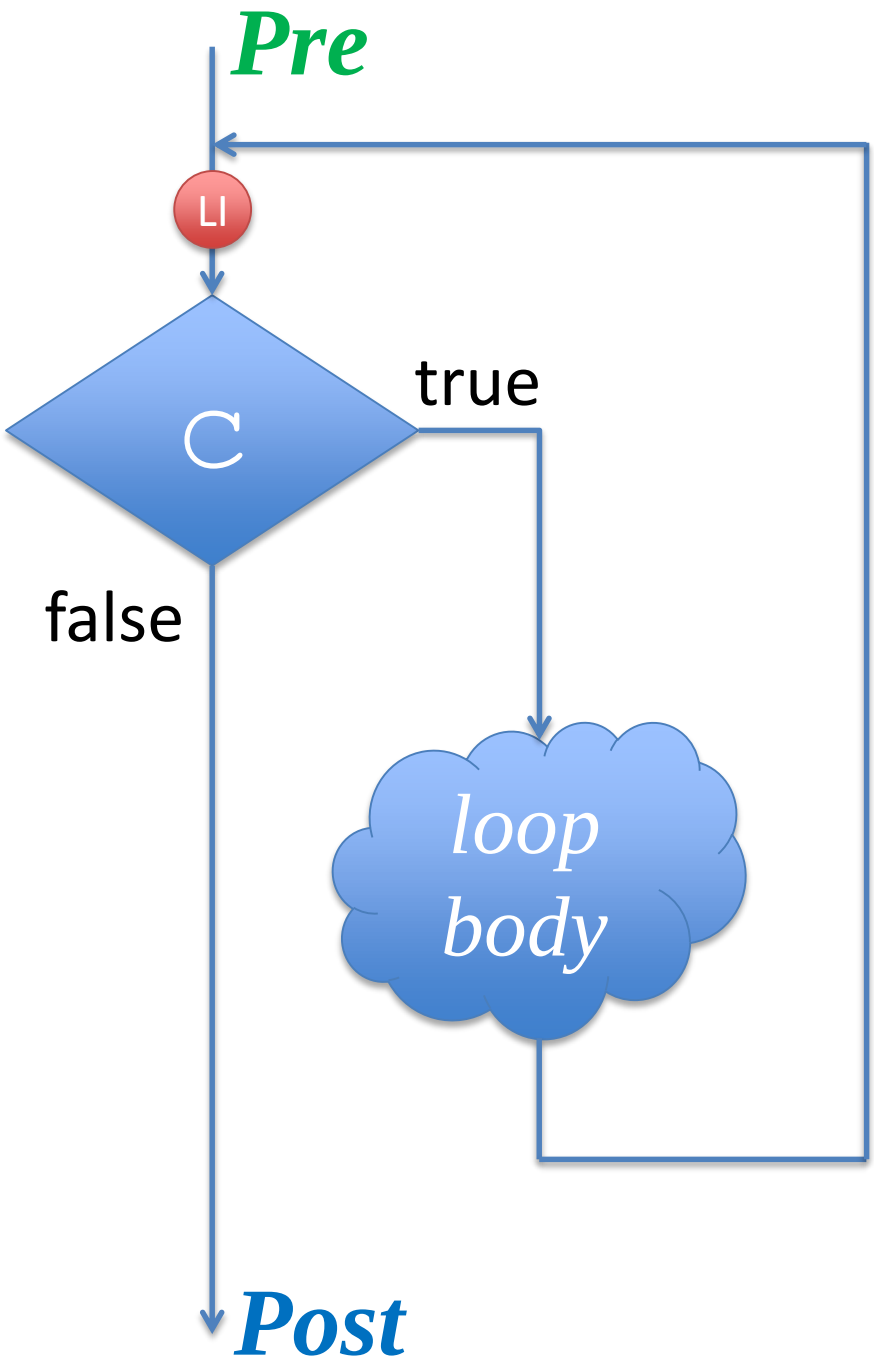
Loop Invariant

A boolean condition that is checked *immediately before every evaluation of the loop guard*.

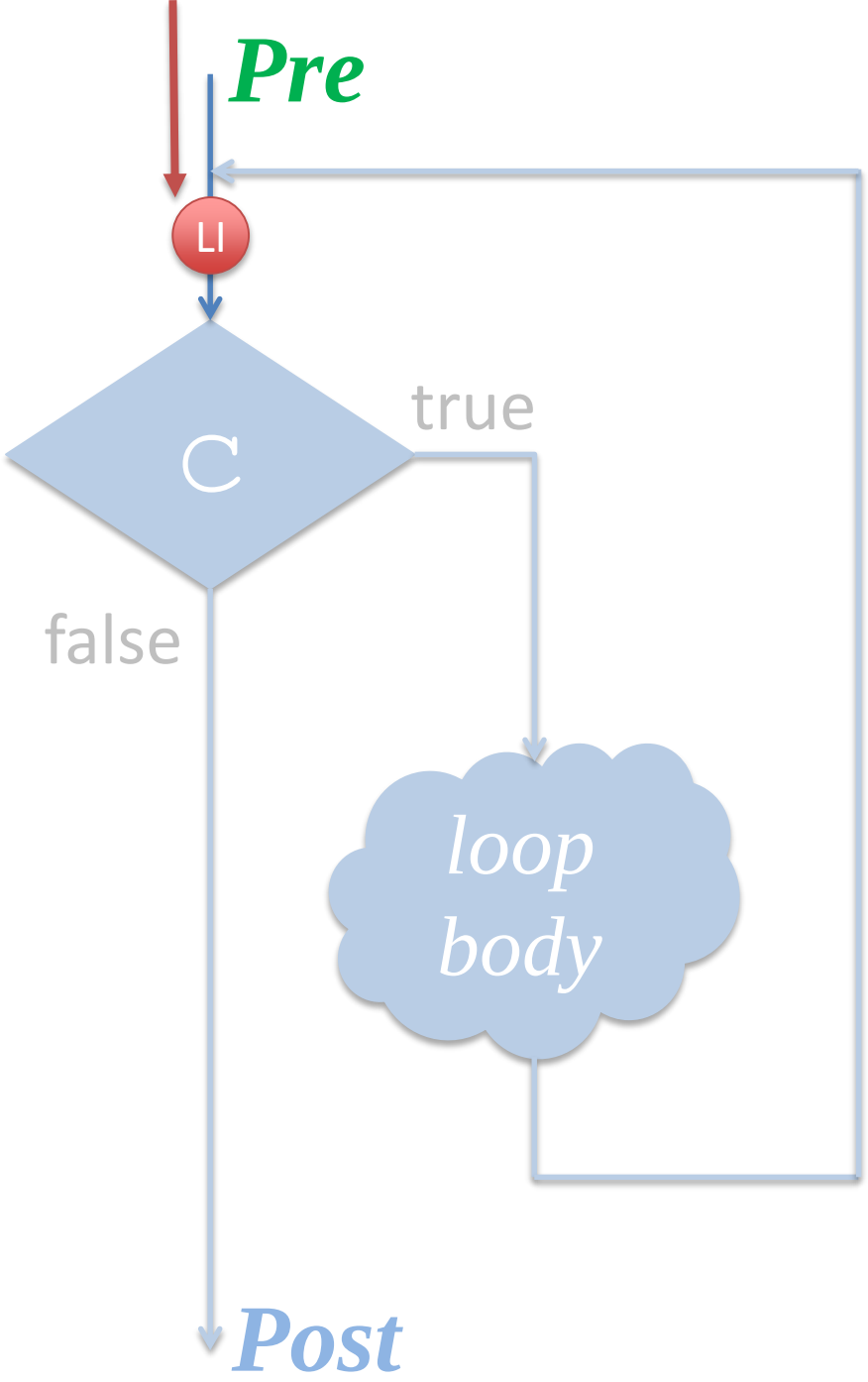
- It is true even if the loop runs 0 times (i.e., is skipped)
- It is true immediately before each evaluation of the loop guard, including the last evaluation if the loop terminates
- It is true immediately after the loop terminates, if the loop terminates

Proving the Correctness
of a function with one loop

Correctness: if preconditions hold,
then postconditions must hold



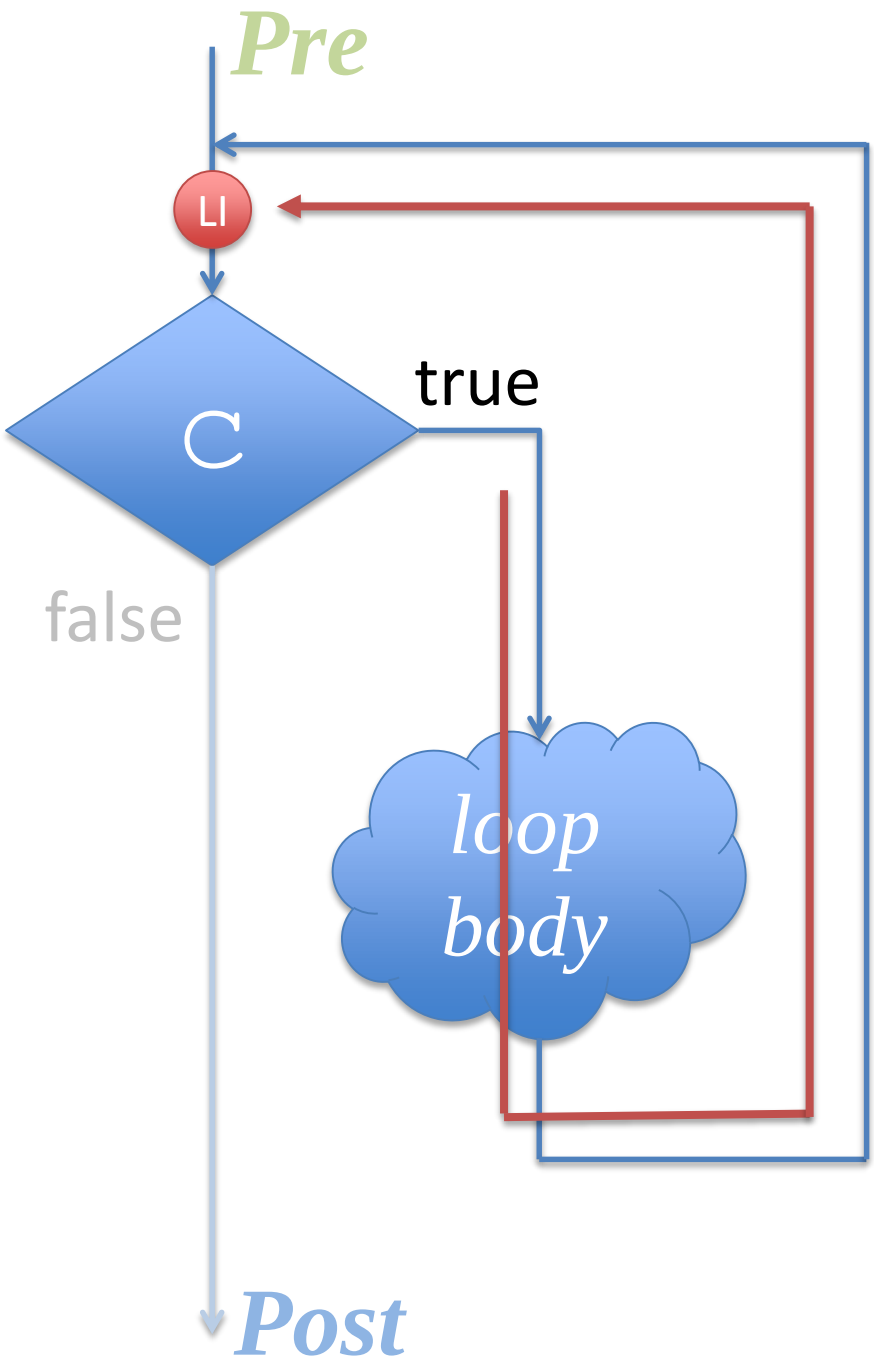

```
//@requires Pre;
//@ensures Post;

   ...

while (C)
//@loop_invariant LI;
{
    loop body
}
```



Showing *LI* valid – 1

INIT

Show that the loop invariant ***LI*** is true immediately before the first evaluation of the loop guard ***C***.



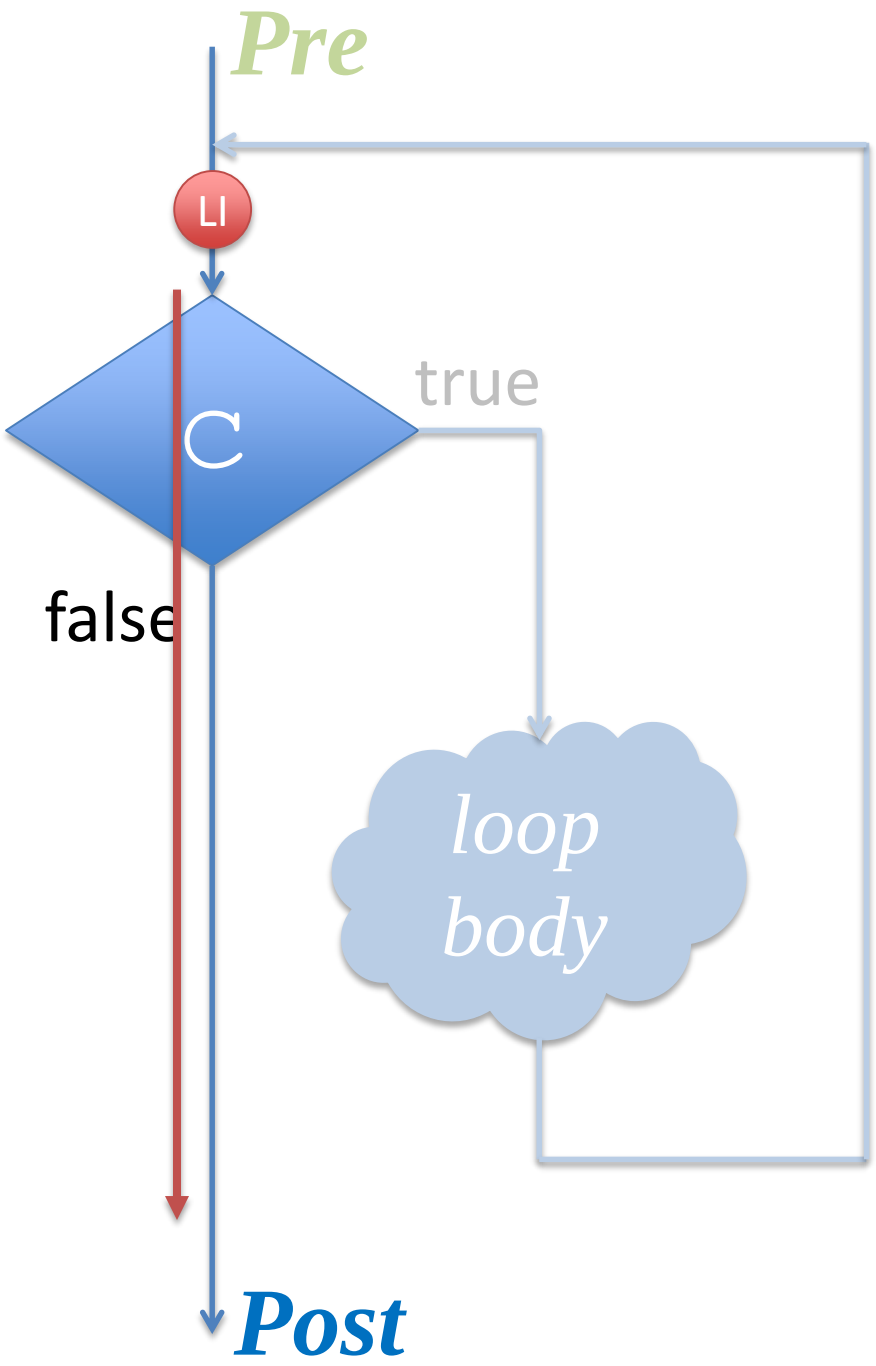
Showing ***LI*** valid – 2

PRESERVATION

Show that:

if the loop invariant ***LI*** is true immediately before the evaluation of the loop guard ***C***,

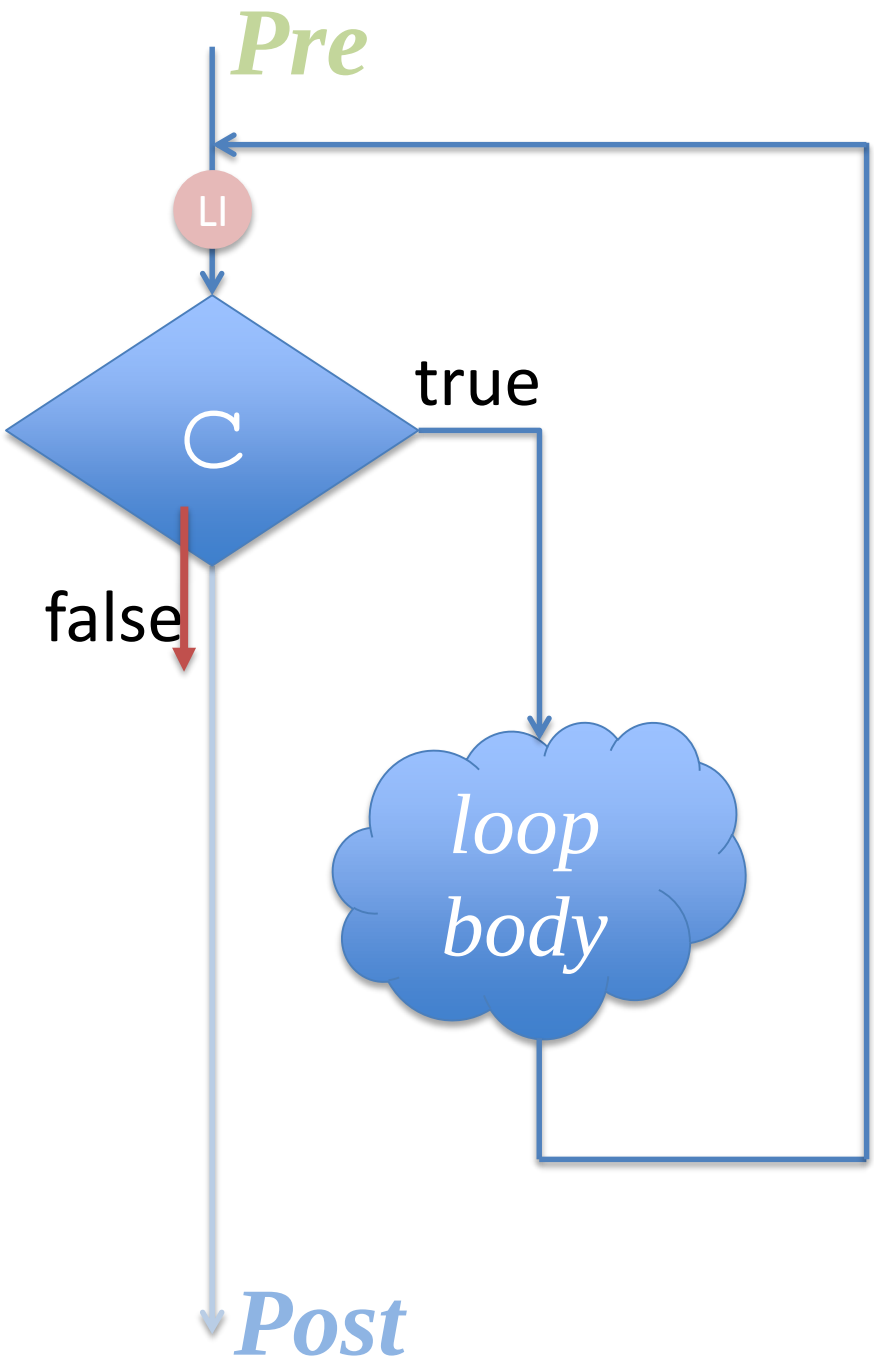
then ***LI*** is true immediately before the next evaluation of the loop guard ***C***.



EXIT

If loop invariant is valid, show that:
the logical conjunction of the loop invariant ***LI*** and the negation of the loop guard ***C*** implies the desired postcondition ***Post***.

$$LI \wedge \sim C \rightarrow Post$$



TERMINATION

Show that the loop will always terminate (i.e., that ***C*** must eventually be false)

Correctness of a function with one loop

- Show that ***LI*** is valid
  - **INIT**: ***LI*** holds initially
  - **PRES**: ***LI*** is preserved by an arbitrary iteration
- **EXIT**: $LI \wedge \sim C \rightarrow Post$
- **TERM**: loop terminates