Prpositions
Predicate
Pre-Conditions
Post-Conditons
Loop:

- Must Terminate
- Must produce the desired output

Loop invariant

- Boolean Statement
  - True at the start of the loop
  - True at the end of each iteration
- Used to prove
  - Properties of loop
  - Partial correctness of loop

EXAMPLE:

[pre-condition: m is a nonnegative integer, x is a real number, $i = 0$, and product = 0.]
while($i \neq m$)
1. product := product + x
2. i := i + 1

end while
[post-condition: product = $m \cdot x$]

PROOF:

Let the loop invariant be
$I(n)$: $i = n$ and product = $n \cdot x$
The guard condition G of the while loop is
G: $i \neq m$

I. Basis Property:
[I(0) is true before the first iteration of the loop.]
$I(0)$: $i = 0$ and product = $0 \cdot x = 0$ Which is true before the first iteration of the loop.

II. Inductive property:
[If the guard G and the loop invariant I(k) are both true before a loop iteration (where $k \geq 0$), then $I(k + 1)$ is true after the loop iteration.]
Before execution of statement 1, productold = $k \cdot x$.
Thus the execution of statement 1 has the effect: productnew = productold + x = $k \cdot x + x = (k + 1) \cdot x$
Similarly, before statement 2 is executed, iold = k,
So after execution of statement 2, inew = iold + 1 = k + 1.
Hence after the loop iteration, the statement $I(k +1)$ (i.e., $i = k + 1$ and product = $(k + 1) \cdot x$) is true.

III. Eventual Falsity of Guard:
[After a finite number of iterations of the loop, the guard becomes false.]

IV. Correctness of the Post-Condition:
[If N is the least number of iterations after which G is false and I(N) is true, then the values of the algorithm variables will be as specified in the post-condition of the loop.]