- Programs or program segments are functions that effect the state of the system

- A computer program:

  1. takes input and assigns names to input data values
  2. manipulates the input, computes new values
  3. produces output, outputs some values

- Statements or groups of statements(program segment) are executed in sequence

- the state of the data values can be examined after the execution of one statement and before execution of the next.

- the program state is the values that are associated with variables of the program

- Consider an example program segment...

```
int n=10;
int i=0;
int sum=0;
while (i!=n){
  sum=sum+i;
  i=i+1;
}
```

- A program segment expresses a partial function e.g.
  i/j cannot be handled for $j = 0$
  no final state corresponds to any initial state where $j = 0$

- Given an initial state, x, of
  $n = 7, i = 4, sum = 2$
  what is the final state of $i = i + 2$?
  Ans: $n = 7, i = 6, sum = 2$

- given the same initial state what is the final state of the program segment, C_1:
  $i = i + 2; sum = sum + i$
  Ans: $n = 7, i = 6, sum = 8$

- Denote by $C_1(x)$ the final state of the system after executing C_1

- C_1, C_2 program segments executed in sequence Concatenated program segment denoted by $C_1; C_2$

- x initial state of C_1. $C_1(x)$ denotes the final state of C_1.

- y denotes initial state of C_2. $C_2(y)$ denotes the final state of C_2

- the final state of $C_1; C_2$ is $C_2(y)$

- the final state of C_1 is the initial state of C_2

- $C_1(x) = y$

- $C_2(y) = C_2(C_1(x))$

- Recall for functions $f : A \to B$ and $g : B \to C$, $g \circ f : A \to C$

- $g \circ f(x) = g(f(x))$ similar to $C_2(C_1(x))$

- $C_2(C_1(x)) = C_2 \circ C_1(x)$

- Note that concatenation is not commutative...see example above

- An assertion is a statement about the program state

- An assertion concerning the initial state is called a precondition. Conditions the input data must satisfy

- An assertion concerning the final state is called the postcondition. What output states are acceptable as correct solutions

- e.g. a program to calculate the real roots of the equation: $ax^2 + bx + c$.

- Roots: $(-b + / - \sqrt(b^2 - 4ac))/2a$.

- Here $b^2 \geq 4ac$ to avoid calculating the square root of a negative number.

- Postcondition: the result returned must be the roots of the equation

- first example precondition: $n \geq 0$ (if $n = 10$ removed)

- Postcondition: $sum = \Sigma_{i=1}^{m} i$ (if $n = 10$ removed)

- (if $n = 10$ included)
  precondition: none or true postcondition: $sum = \Sigma_{i=1}^{10} i$

- The precondition and postcondition describe what the program has to achieve

- The pre and postcondition are boolean valued functions

- A program segment is correct if all states satisfying the precondition, will lead to states satisfying the postcondition