

```
while( n < (document.
{
    n++;
    calc = ev
    i++
    i++
```



Programming Language

Example: C

Sequence

- Code consists of a sequence of statements (and comments)
 - These are executed sequentially, one after the other
- Generally, code is evaluated statement by statement, left to right
 - E.g.: First compute x , then compute $(x+1)$, then multiply result by x and assign it to variable y
- Basic mathematics can be expressed directly
- Examples:
 $x = 5 + 5;$ // compute $5 + 5$ then assign the result to the variable x
 $y = (x + 1) * x;$ // use the value of x from the previous line!

The if Conditional Statement

- `if` checks the value of an expression and runs the statement if true

`if (<expression>) <statement>`

- Use the following notation:

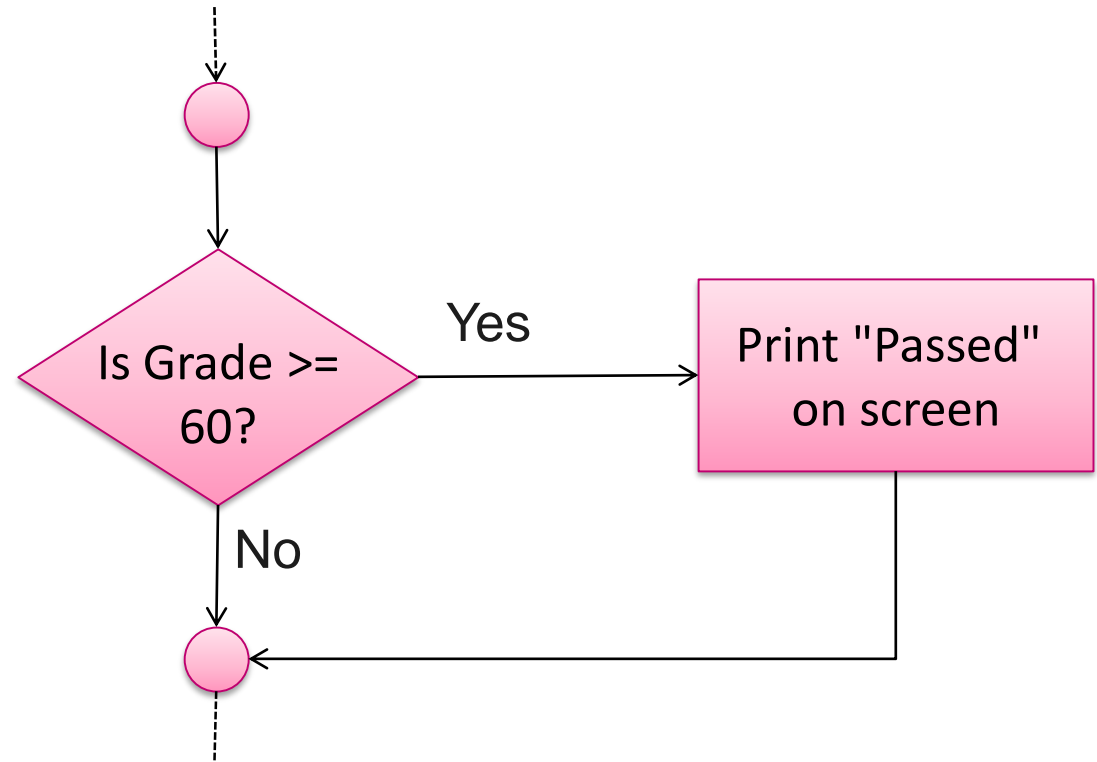
```
if( condition ){  
    // Code to be executed if the condition is true  
}
```

- If the condition is TRUE, i.e. `!= 0`, then execute the statement
- If the condition is FALSE, then the statement is ignored

The if Conditional Statement

- Example code:

```
if( grade >= 60 ){  
    printf("Passed");  
}
```



if...else conditional statement

- `if` may be supplemented by the `else` command.
- Use the `else` statement to indicate what to do when the condition is FALSE (0)
- Example:

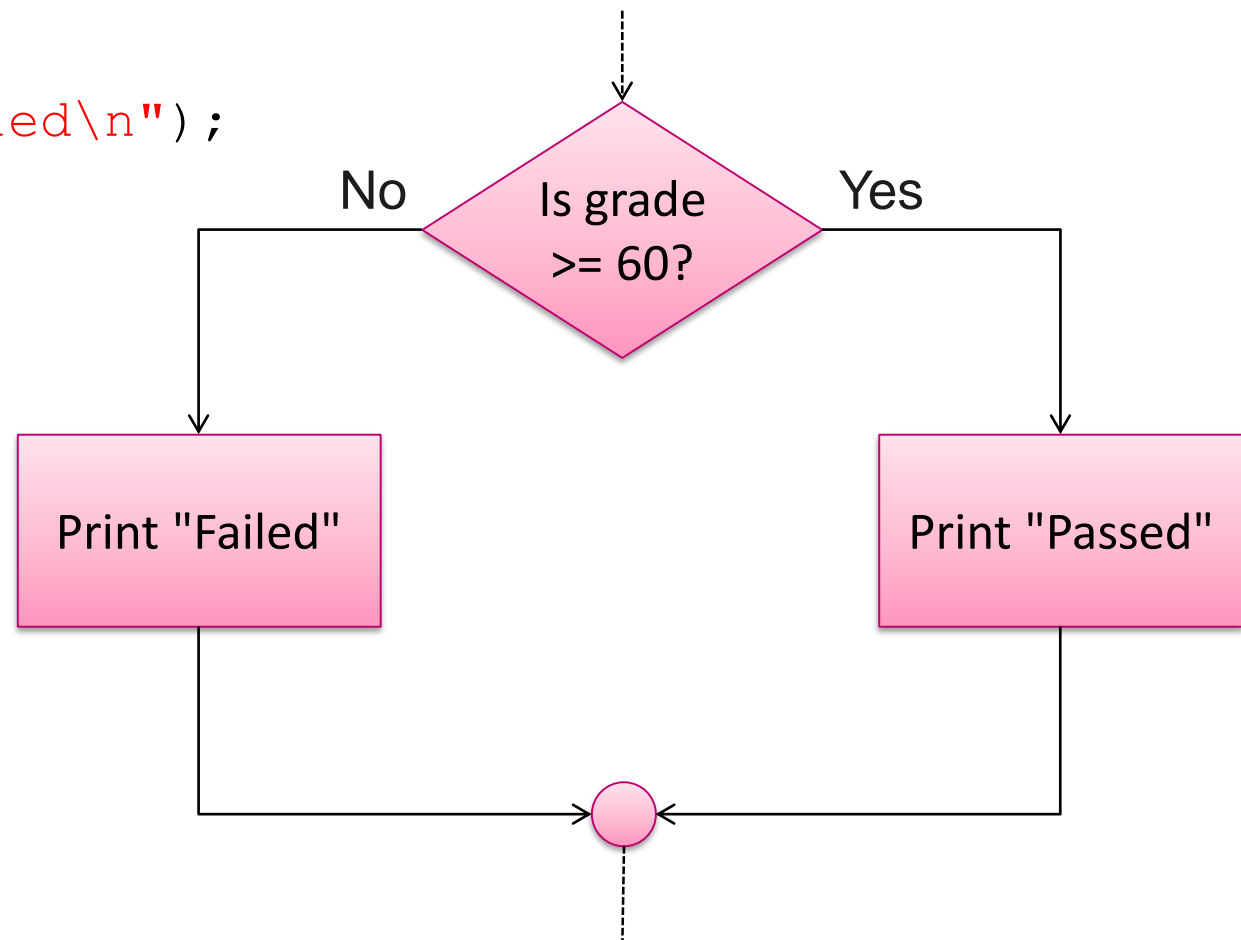
```
if( condition ){  
    // Code is executed when condition is true  
}else{  
    // Code is executed when condition is false  
}
```

Don't need { and } for just one line of code, however recommended for clarity

if...else conditional statement

- In C code:

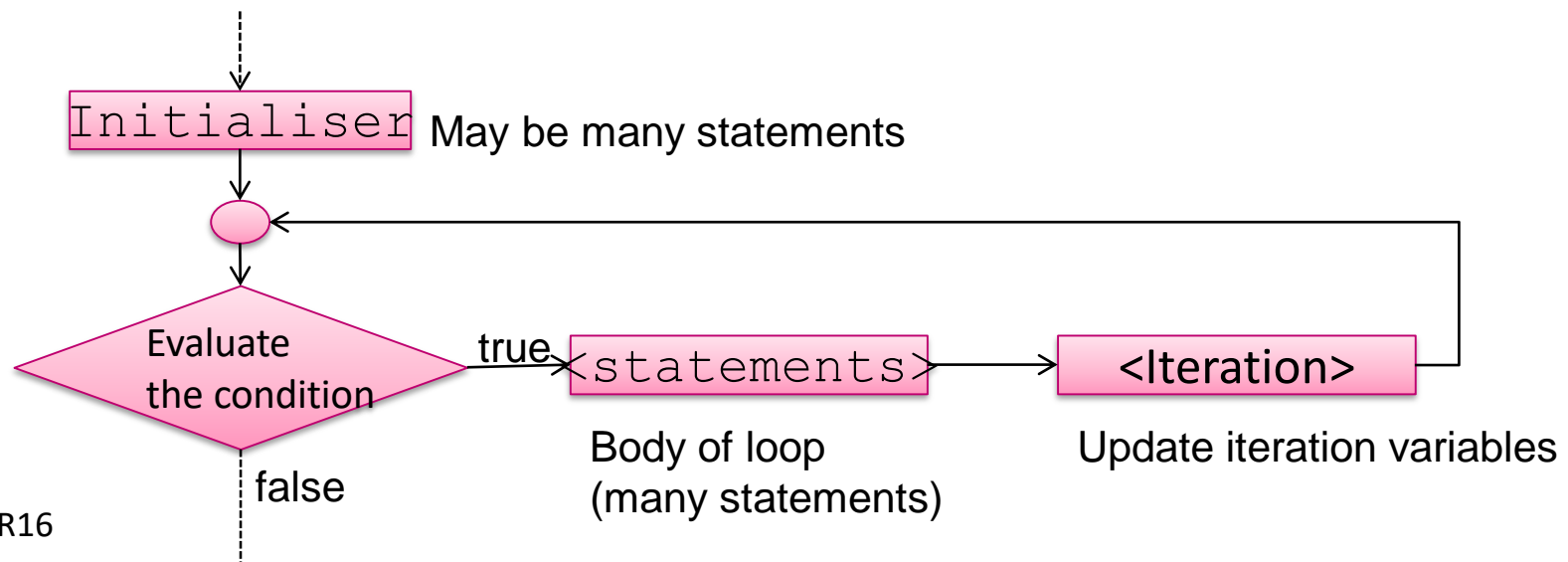
```
if( grade >= 60 ) {  
    printf("Passed\n");  
}else{  
    printf("Failed\n");  
}
```



Iteration: The For Loop

- The for loop is optimal for counting or "iterating"
 - It allows a group of instructions to be executed a given number of times
- Key Points:
 - Has an innate ability to handle counters

```
for( <initialiser> ; <condition> ; <iteration> ){  
    <statements>  
}
```
 - If the loop condition is initially FALSE, then the body is not executed



For Loop: Example

- Iterating over code 10 times

```
int counter;
```

Counter is the control variable

Initialise
counter

Counter
limit

Increment
command

```
for(counter=1; counter<=10; counter++) {  
    printf("%d\n", counter);  
}
```

- This code prints the numbers 1 to 10