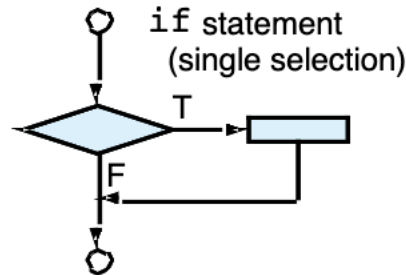


# Structured-Programming Summary

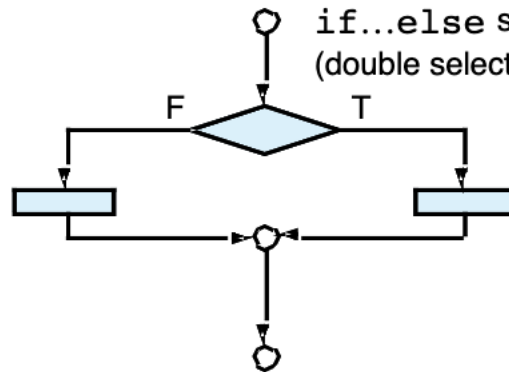
Sequence



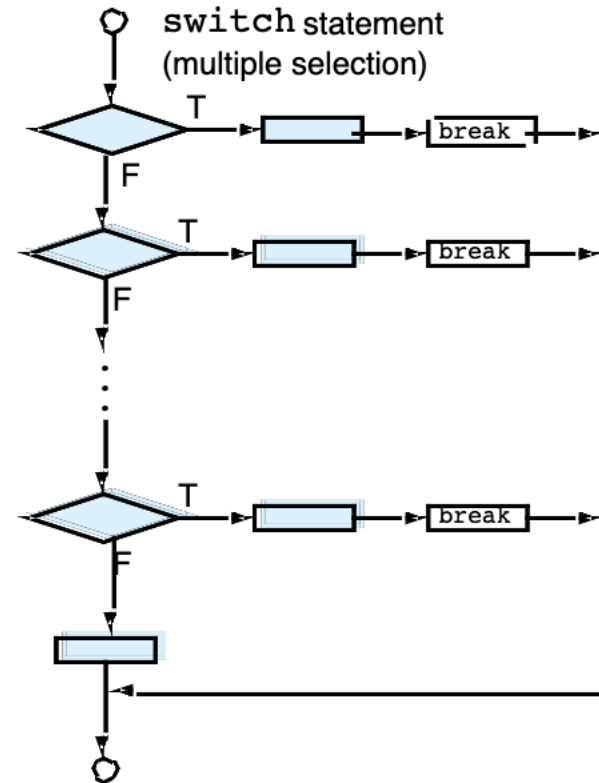
if statement  
(single selection)



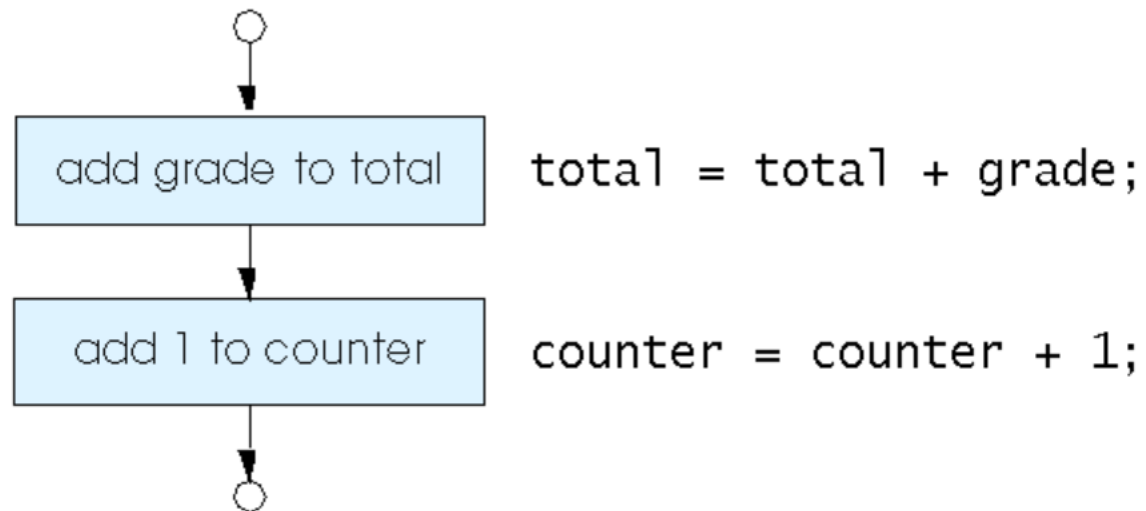
if...else statement  
(double selection)



switch statement  
(multiple selection)



Flowcharting C's sequence structure.

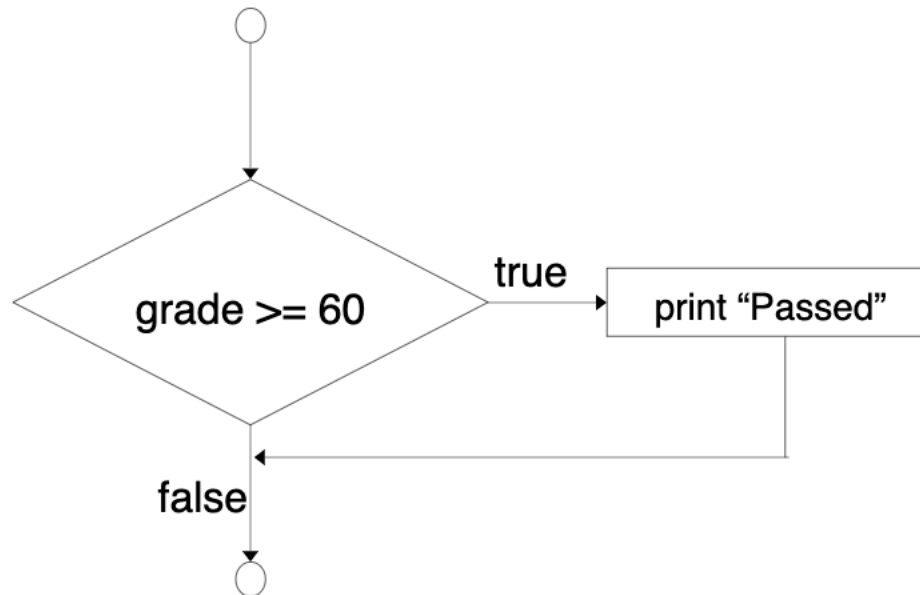


# The `if` Selection Statement

- Pseudocode statement in C:

```
if ( grade >= 60 )  
    printf( "Passed\n" );
```

  - C code corresponds closely to the pseudocode
- `if` statement is a single-entry/single-exit structure



A decision can be made on any expression.

zero - false

nonzero - true

Example:

3 - 4 is true

# The `if...else` Selection Statement

- `if...else`
  - Specifies an action to be performed both when the condition is `true` and when it is `false`
- C code:

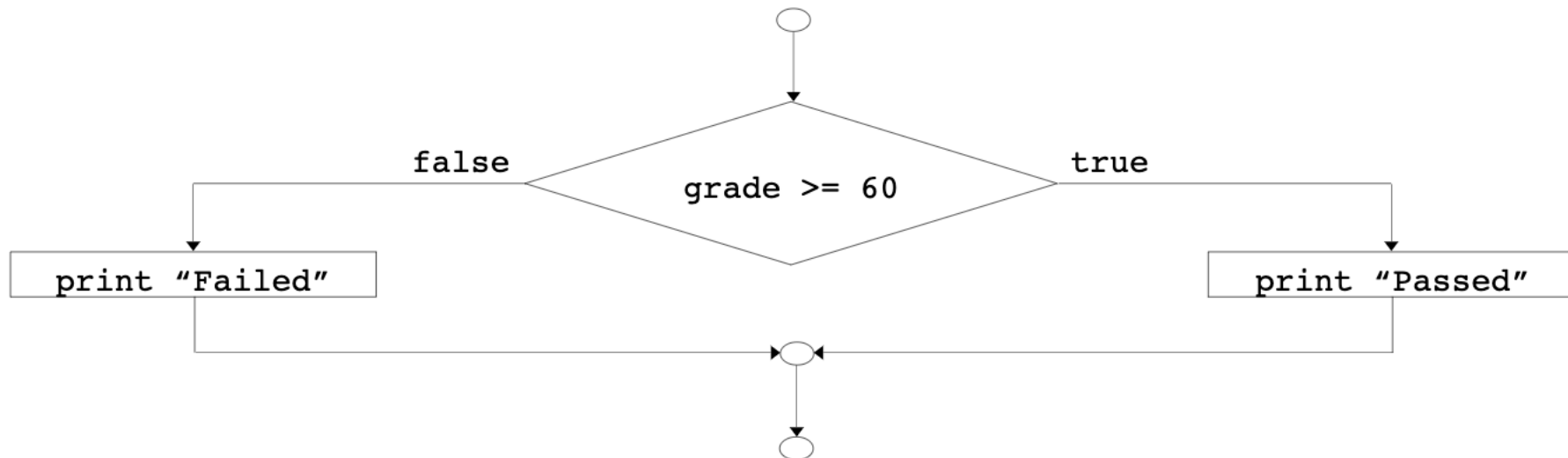
```
if ( grade >= 60 )  
    printf( "Passed\n" );  
else  
    printf( "Failed\n" );
```
- Ternary conditional operator (`?:`)
  - Takes three arguments (condition, value if `true`, value if `false`)
  - Our pseudocode could be written:

```
printf( "%s\n", grade >= 60 ? "Passed" : "Failed" );
```
  - Or it could have been written:

```
grade >= 60 ? printf( "Passed\n" ) :  
    printf( "Failed\n" );
```

# The if...else Selection Statement

- Flow chart of the if...else selection statement



- Nested if...else statements
  - Test for multiple cases by placing if...else selection statements inside if...else selection statement
  - Once condition is met, rest of statements skipped
  - Deep indentation usually not used in practice

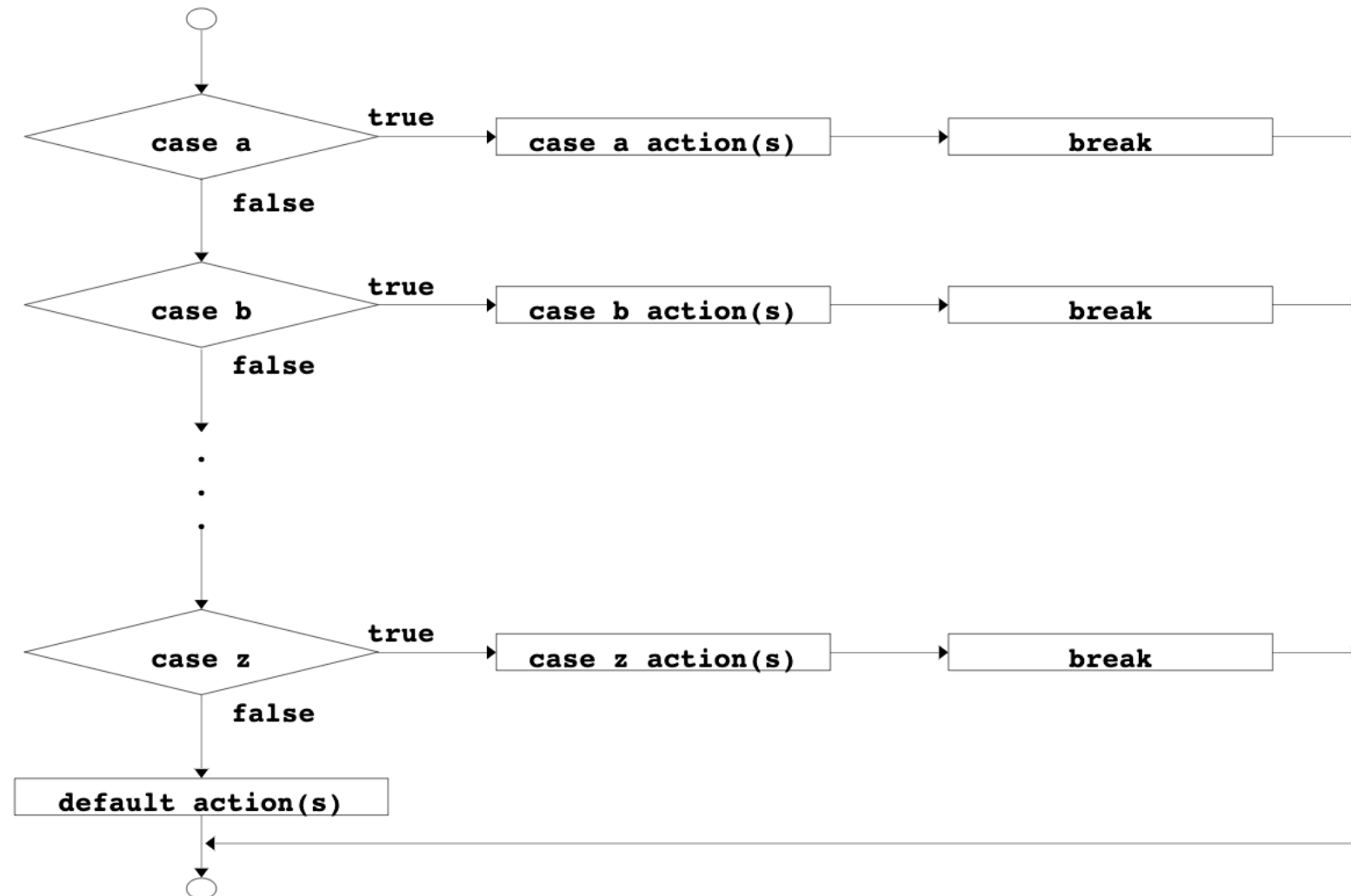
# The switch Multiple-Selection Statement

- `switch`
  - Useful when a variable or expression is tested for all the values it can assume and different actions are taken
- Format
  - Series of case labels and an optional default case

```
switch ( value ){
    case '1':
        actions
    case '2':
        actions
    default:
        actions
}
```
  - `break;` exits from statement

# The switch Multiple-Selection Statement

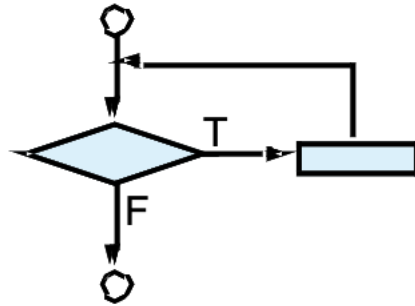
- Flowchart of the switch statement



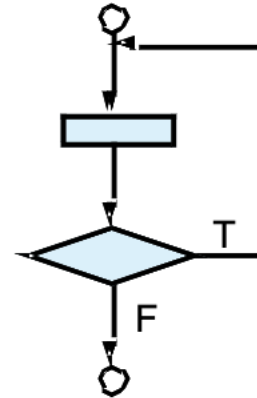
# Structured-Programming Summary

## Repetition

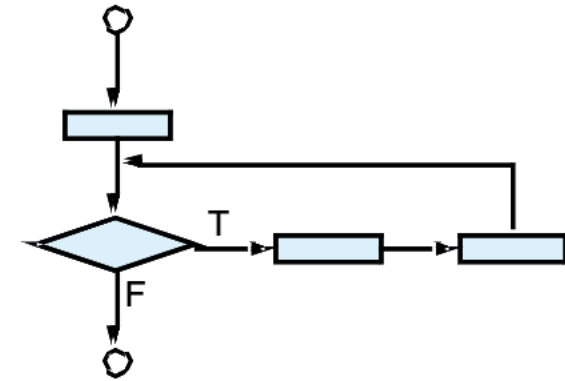
while statement



do..while statement



for statement

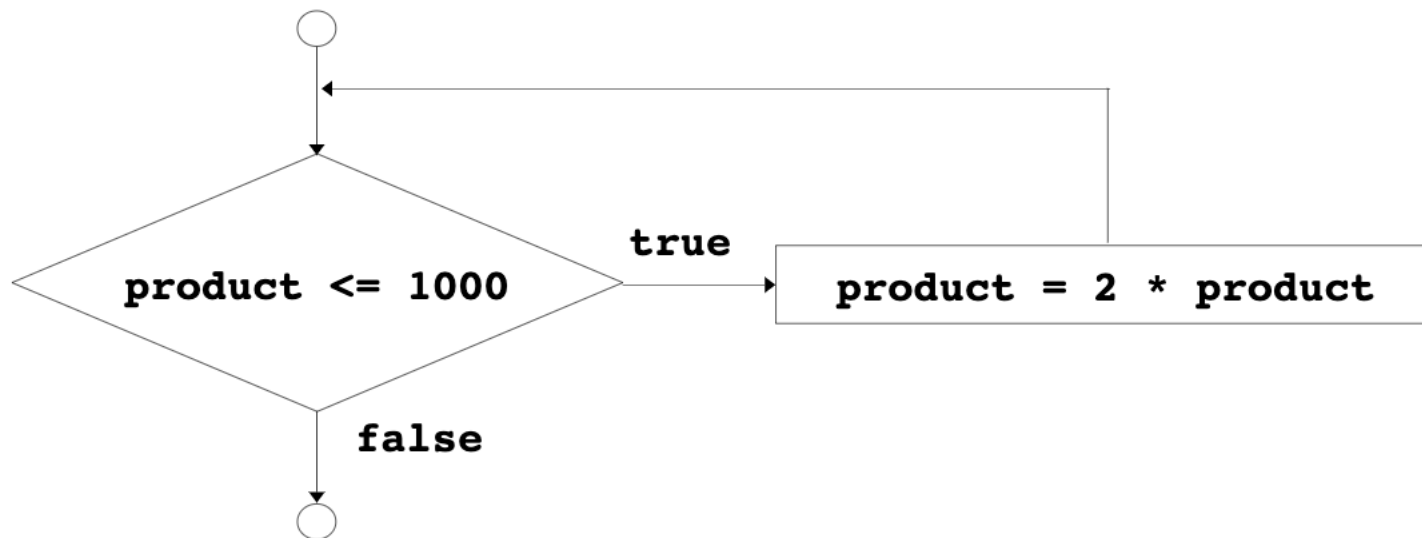




# The while Repetition Statement

- Example:

```
int product = 2;  
while ( product <= 1000 )  
    product = 2 * product;
```



# The do...while Repetition Statement

- The do...while repetition statement
  - Similar to the while structure
  - Condition for repetition tested after the body of the loop is performed
    - All actions are performed at least once
  - Format:

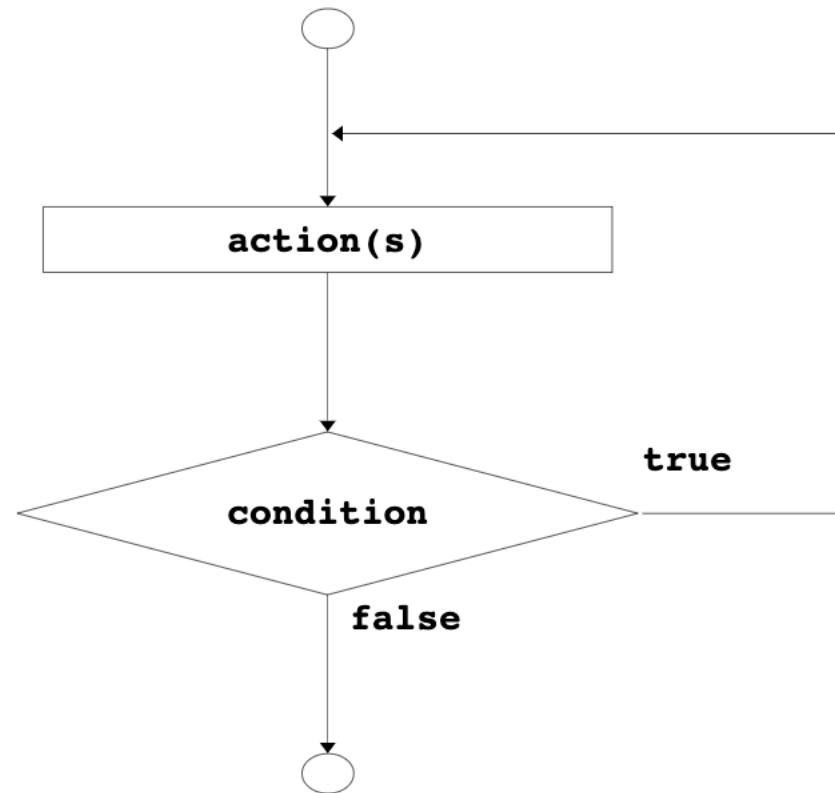
```
do {  
    statement;  
} while ( condition );
```
- Example (letting counter = 1):

```
do {  
    printf( "%d  ", counter );  
} while (++counter <= 10);
```

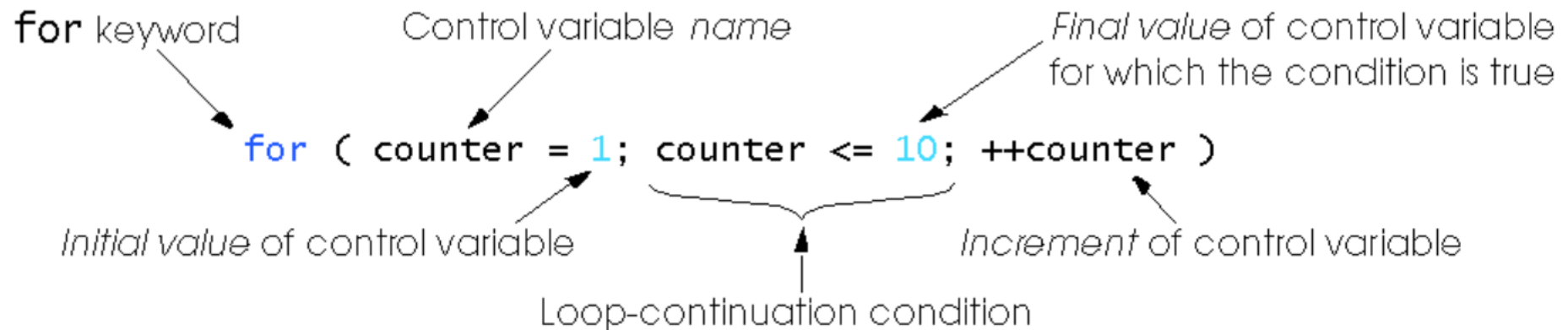
  - Prints the integers from 1 to 10

# The do...while Repetition Statement

- Flowchart of the do...while repetition statement



# The for Repetition Statement



Format when using for loops

```
for ( initialization; loopContinuationTest; increment )  
    statement
```

Example:

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

Prints the integers from one to ten

No  
semicolon  
(;) after last  
expression