

Breaking the flow: The if statement

Computer Programming 143 – Lecture 3

Structured Program Development I

Faculty of Engineering



3.4 Control Structures I

All programs written in terms of 3 control structures

- **Sequence structures:** Built into C. Programs executed sequentially by default
- **Selection structures:** C has three types: `if`, `if...else`, and `switch`
- **Repetition structures:** C has three types: `while`, `do...while`, and `for`

Decisions

Loops

Algorithms

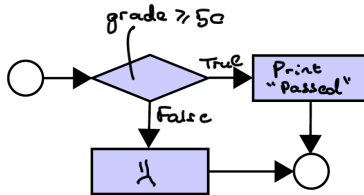
- Algorithm: Recipe or steps
- Program: Actual instructions following correct syntax (in C)

Pseudo-code

PASS / FAIL PROGRAM

```
Read in student's grade
if grade greater than 50%:
    | print "Passed!"
otherwise:
    | print "Failed!"
    | print encouragement
```

3.4 Flow diagrams

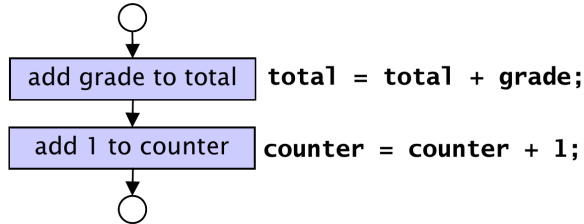


- Graphical representation of an algorithm
- Drawn using certain special-purpose symbols connected by arrows called flow lines
- Symbols:
 - Rectangle: Indicates any type of action
 - Oval: Indicates the beginning or end of a program or a section of code
 - Diamond: Indicates decision is to be made

3.4 Control Structures III

Control Structure 1: Sequential execution

- Statements executed one after the other in the order they were written



Back to our pass-fail example

```
read in a student's grade
if grade greater than 50:
    print "passed"
otherwise:
    print "failed"
    print some encouragement
```

scanf ✓

if ... else statement

printf ✓

The diagram illustrates the flow of a pass-fail program. It starts with 'read in a student's grade', which is annotated with a handwritten arrow from 'scanf' and a blue checkmark. This leads to an 'if' statement 'if grade greater than 50:'. The 'if' branch contains 'print "passed"', annotated with a handwritten arrow from 'if ... else statement'. The 'otherwise' branch contains 'print "failed"' and 'print some encouragement', with a handwritten arrow from the same 'if ... else statement' annotation pointing to the 'otherwise:' label. The 'print' statements are annotated with a handwritten arrow from 'printf' and a blue checkmark.

The if statement

```
if ( condition )  
{  
    // what happens if condition is true  
    ;  
}  
else  
{  
    // happens if condition is false  
}  
}
```

3.5 The 'if' Selection Statement II

C code:

```
if ( grade >= 50 ) {  
    printf( "Passed\n" );  
}
```

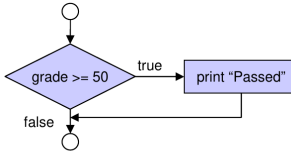
- C code corresponds closely to the pseudocode
- A set of braces may be removed when it contains only one statement

C code:

```
if ( grade >= 50 )  
    printf( "Passed\n" );
```

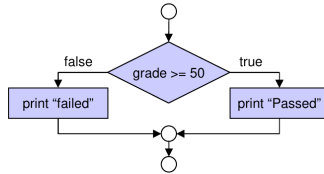
3.5 The 'if' Selection Statement III

- **if** statement is a single-entry/single-exit structure



3.6 The 'if...else' Statement II

Flow diagram of the 'if...else' selection statement



2.6 Decision Making I

Equality operators

- == Is equal to
- != Is not equal to

Relational operators

- > Greater than
- < Less than
- >= Greater than or equal to
- <= Less than or equal to

3.6 The 'if...else' Statement III

Pseudocode:

```
If grade is greater than or equal to 80
{
    Print "A"
}
else
{
    If grade is greater than or equal to 70
    {
        Print "B"
    }
    else
    {
        If grade is greater than or equal to 60
        {
            Print "C"
        }
        else
        {
            If grade is greater than or equal to 50
            {
                Print "D"
            }
        }
    }
}
```

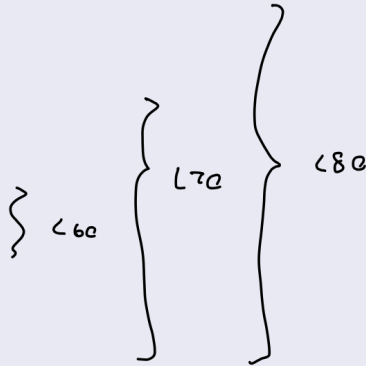
$70 \leq \text{grade} < 80$

< 70

3.6 The 'if...else' Statement IV

C code

```
if ( grade >= 80 ) {  
    printf( "A\n" );  
} // end if  
else {  
    if ( grade >= 70 ) {  
        printf( "B\n" );  
    } // end if  
    else {  
        if ( grade >= 60 ) {  
            printf( "C\n" );  
        } // end if  
        else {  
            if ( grade >= 50 ) {  
                printf( "D\n" );  
            } // end if  
        } // end else  
    } // end else  
} // end else
```



3.6 The 'if...else' Statement V

Testing multiple conditions

- The following 'if' statement is *wrong*:

```
if ( 0 < x < 5 )  
    printf( "x lies between 0 and 5" );
```

- Use nested 'if' statements (for now):

```
if ( x > 0 )  
    if ( x < 5 )  
        printf( "x lies between 0 and 5" );
```

Homework

Question: Write a program that reads in two numbers and tells you whether they are the same or which one is bigger (if they are different).

Memo: Slides 16-17