

Decisions with Variables

- Need to take *logical decisions during problem solving*
 - If $b^2 - 4ac$ negative, we should report that the quadratic equation has no real roots
- The *if-else* programming construct provides the facility to make logical decisions

The if-else construct

if (condition)

{

evaluate this part if cond is true

}

else // Optional

{

evaluate this part if false

}

If Statement

if (<expression>) <stmt1> [*else* <stmt2>]

if (<expression>) {<stmt1>} [*else* {<stmt2>}]

Semantics:

Expression evaluates to “true”

- stmt1 will be executed

Expression evaluates to “false”

- stmt2 will be executed

Else part is optional

Expression is “true” -- stmt1 is executed; else, no action

Conditions

- Specified using relational and equality operators
- Relational: $>$, $<$, $>=$, $<=$
- Equality: $==$, $!=$
- Usage: for a, b values or variables
 $a > b$, $a < b$, $a >= b$, $a <= b$, $a == b$, $a != b$
- A condition is satisfied or true, if the relational operator, or equality is satisfied.
- For $a = 3$, and $b = 5$:
 - $a < b$, $a <= b$, and $a != b$ are true
 - $a > b$, $a >= b$, $a == b$ are false

Selection Statements

Three forms:

single selection:

if (att < 85) grade = 'W';

no *then* reserved
word

double selection:

if (marks < 40)

passed = 0;

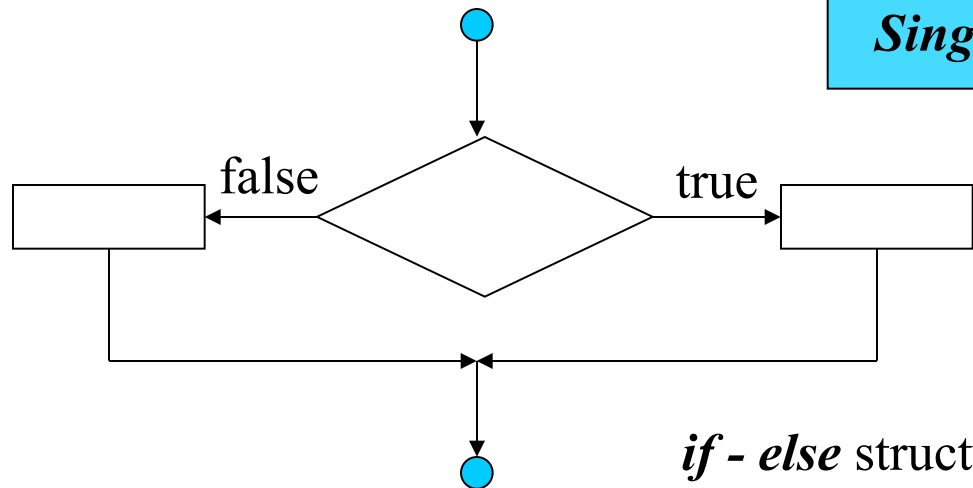
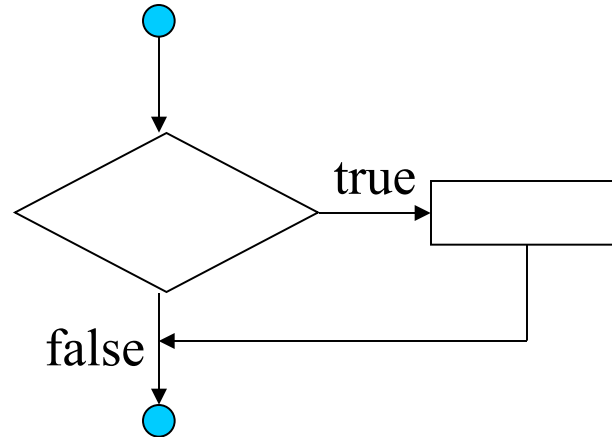
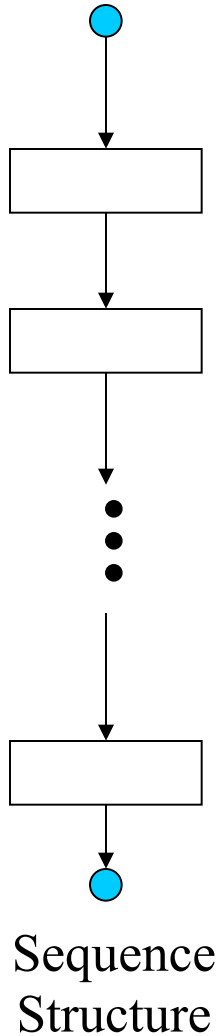
else

passed = 1;

multiple selection:

switch statement - to be discussed later

Sequence and Selection Flowcharts



If structure

***Single Entry
Single Exit***

Grading Example

Below 50: D; 50 to 59: C; 60 to 75: B; 75 above: A

int marks;

char grade;

...

if (marks < 50) grade = 'D';

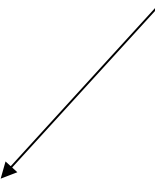
else if (marks <= 59) grade = 'C';

else if (marks <=75) grade = 'B';

else grade = 'A';

...

Note the semicolon
before else !



Unless braces are used, an else part
goes with the nearest else-less if stmt

Grading Example, using {}

```
if (marks < 50)
{ grade = 'D'; }
else
{
    if (marks <= 59) { grade = 'C'; }
    else
    {
        if (marks <= 75) grade = 'B';
        else grade = 'A';
    }
}
```

Grading Example 2

Below 50: D; 50 to 59: C ; 60 to 75: B; 75 above: A

int marks;

char grade;

...

if (marks>75) grade='A';

if ((marks>=60) && (marks<=75)) grade='B';

if ((marks>=50) && (marks<=59)) grade='C';

Objective 1

- Marks ≥ 40 -> Passed
- Marks < 40 -> Failed

```
if (marks  $\geq$  40)  
    printf(“you passed 8-”);  
else  
    printf(“you failed 8-”);
```

- Objective 1 is changed to **add**
 - If Marks > 75 , declare Distinction
 - No need to mention Passed

Caution in use of “else”

```
if ( marks > 40)
```

/ WRONG – else goes
with second if */*

```
    if ( marks > 75 ) printf(“you got distinction”);  
    else printf(“you failed”);
```

```
if ( marks > 40) {
```

*/*RIGHT*/*

```
    if ( marks > 75 ) printf(“you got distinction”);  
    else printf(“you passed\n”);  
}
```

```
else printf(“you failed”);
```