

COMP 141

if-elif-else, and/or



1

Announcements

- Reminders:
 - Program 1 due tomorrow night by 11:55pm

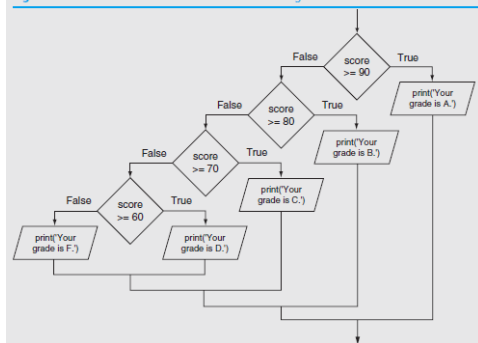


2

Practice

Write a program that prompts a user for his or her age and prints out whether or not they are (legally) allowed to drink alcohol.

Figure 4-17 Nested decision structure to determine a grade



4

```
# This program gets a numeric test score from the
# user and displays the corresponding letter grade.
```

```
# Constants for the grade thresholds
```

```
A_SCORE = 90
```

```
B_SCORE = 80
```

```
C_SCORE = 70
```

```
D_SCORE = 60
```

```
# Get a test score from the user.
```

```
score = int(input('Enter your test score: '))
```

```
# Determine the grade.
```

```
if score >= A_SCORE:
```

```
    print('Your grade is A.')
```

```
else:
```

```
    if score >= B_SCORE:
```

```
        print('Your grade is B.')
```

```
    else:
```

```
        if score >= C_SCORE:
```

```
            print('Your grade is C.')
```

```
        else:
```

```
            if score >= D_SCORE:
```

```
                print('Your grade is D.')
```

```
            else:
```

```
                print('Your grade is F.')
```



5

The if-elif-else Statement

- **if-elif-else statement:** special version of a decision structure

- Makes logic of nested decision structures simpler to write

- Can include multiple elif statements

- Syntax:


```
if condition1:
    statements
elif condition2:
    statements
else:
    statements
```



6

```
# This program gets a numeric test score from the
# user and displays the corresponding letter grade.
# It is equivalent to the program in the previous slide
```

```
# Constants for the grade thresholds
```

```
A_SCORE = 90
```

```
B_SCORE = 80
```

```
C_SCORE = 70
```

```
D_SCORE = 60
```

```
# Get a test score from the user.
```

```
score = int(input('Enter your test score: '))
```

```
# Determine the grade.
```

```
if score >= A_SCORE:
```

```
    print('Your grade is A.')
```

```
elif score >= B_SCORE:
```

```
    print('Your grade is B.')
```

```
elif score >= C_SCORE:
```

```
    print('Your grade is C.')
```

```
elif score >= D_SCORE:
```

```
    print('Your grade is D.')
```

```
else:
```

```
    print('Your grade is F.')
```



7

Logical Operators

- **Logical operators:** operators that can be used to create complex Boolean expressions

- **and** operator and **or** operator: binary operators, connect two Boolean expressions into a compound Boolean expression

- **not** operator: unary operator, reverses the truth of its Boolean operand



8

The and Operator

```
if ____ and ____ :
    # do something
else:
    # do something else
```

Both individual tests must be **True** to make the entire if statement **True**.

Truth table for the and operator

Expression	Value of the Expression
false and false	false
false and true	false
true and false	false
true and true	true

The or Operator

```
if ____ or ____ :
    # do something
else:
    # do something else
```

Either (or both) individual tests must be **True** to make the entire if statement **True**.

Truth table for the or operator

Expression	Value of the Expression
false and false	false
false and true	true
true and false	true
true and true	true

Short-Circuit Evaluation

- **Short circuit evaluation:** deciding the value of a compound Boolean expression after evaluating only one sub expression
 - Performed by the **or** and **and** operators
 - For **or** operator: If left operand is true, compound expression is true. Otherwise, evaluate right operand
 - For **and** operator: If left operand is false, compound expression is false. Otherwise, evaluate right operand

The not Operator

- Takes a Boolean expression as operand and reverses its logical value
 - Sometimes it may be necessary to place parentheses around an expression to clarify to what you are applying the not operator

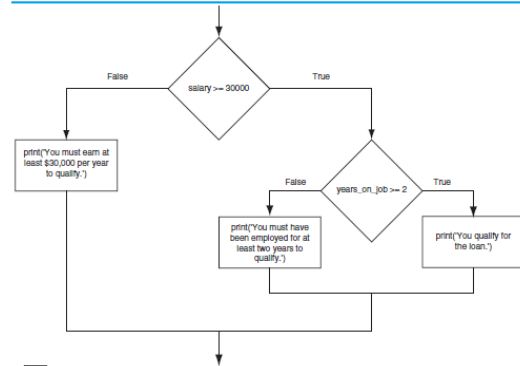
Truth table for the not operator

Expression	Value of the Expression
true	false
false	true

Checking Numeric Ranges with Logical Operators

- To determine whether a numeric value is within a specific range of values, use **and**
 - Example: $x \geq 10$ and $x \leq 20$
- To determine whether a numeric value is outside of a specific range of values, use **or**
 - Example: $x < 10$ or $x > 20$

Figure 4-14 A nested decision structure



```

# This program determines whether a bank customer
# qualifies for a loan.

# Constants for minimum salary and minimum
# years on the job
MIN_SALARY = 30000.0
MIN_YEARS = 2

# Get the customer's annual salary.
salary = float(input('Enter your annual salary: '))

# Get the number of years on the current job.
years_on_job = int(input('Enter the number of ' +
    'years employed: '))

# Determine whether the customer qualifies.
if salary >= MIN_SALARY:
    if years_on_job >= MIN_YEARS:
        print('You qualify for the loan.')
    else:
        print('You must have been employed', \
            'for at least', MIN_YEARS, \
            'years to qualify.')
else:
    print('You must earn at least $', \
        format(MIN_SALARY, ',.2f'), \
        'per year to qualify.', sep='')
  
```

```

# This program determines whether a bank customer
# qualifies for a loan.

# Constants for minimum salary and minimum
# years on the job
MIN_SALARY = 30000.0
MIN_YEARS = 2

# Get the customer's annual salary.
salary = float(input('Enter your annual salary: '))

# Get the number of years on the current job.
years_on_job = int(input('Enter the number of ' +
    'years employed: '))

# Determine whether the customer qualifies.
if salary >= MIN_SALARY and years_on_job >= MIN_YEARS:
    print('You qualify for the loan.')
else:
    print('You do not qualify for this loan.')
  
```

Review Questions

1. Does an `if` statement **always** need to be followed by an `else` statement?
2. If you write an `if-else` statement, under what circumstances do the statements that appear after the `else` clause execute?
3. Assume the variables `a = 2`, `b = 4`, `c = 6`. What do the following statements evaluate to (true or false)?
 - a) `a == 4 or b > 2`
 - b) `6 <= c and a > 3`
 - c) `1 != b and c != 3`
 - d) `a >= -1 or a <= b`
 - e) `not (a > 2)`

In-Class Lab