

Objectives

- More conditionals
- Boolean operators

Vehicle Information

undefined

VA NotApplicable

ADD VEHICLE

Review

- How can we make Python code execute only under certain circumstances?
- How do we say “otherwise” in Python?
- How do we write the condition that evaluates to True if two expressions (let’s say `expr1` and `expr2`) are equal?
 - How do we write the condition to evaluate to True only if those two expressions are *not* equal?

Oct 16, 2017

Sprenkle - CSCI111

3

Review: Syntax of `if` statement: Simple Decision

```
if condition :  
    statement1  
    statement2  
    ...  
    statementn
```

keyword

“then” Body

- Note indentation

English Examples:

```
if it is raining :  
    I will wear a raincoat  
if the PB is new :  
    Remove the seal
```

Oct 16, 2017

Sprenkle - CSCI111

4

Syntax of `if` statement:

Two-Way Decision

English Example:

```
if condition :  
    statement1  
    statement2  
    ...  
    statementn  
else :  
    statement1  
    statement2  
    ...  
    statementn
```

“then” Body

“else” Body

keywords

if it is Saturday or Sunday :
 I wake up at 10 a.m.
else :
 I wake up at 7 a.m.

Oct 16, 2017

Sprenkle - CSCI111

5

Review: Relational Operators

- Syntax:

➤ `<expr> <relational_operator> <expr>`

| | Relational Operator | Meaning |
|-----------------------|---------------------|---------------------------|
| Low precedence | < | Less than? |
| | <= | Less than or equal to? |
| | > | Greater than? |
| | >= | Greater than or equal to? |
| | == | Equals? |
| | != | Not equals? |

Oct 16, 2017

Sprenkle - CSCI111

6

Review: Using Conditionals

- Determine if a number is even or odd

```
x = eval(input("Enter a number: "))
remainder = x%2
if remainder == 0:
    print(x, "is even")
if remainder == 1:
    print(x, "is odd")
```

```
x = eval(input("Enter a number: "))
remainder = x % 2
if remainder == 0:
    print(x, "is even")
else:
    print(x, "is odd")
```

This is the more efficient implementation. Why?

Oct 16, 2017

Speeding Ticket Fine

```
# getting the necessary input from the user
speed = eval(input("Enter your speed: "))
speedlimit = eval(input("Enter the speed limit: "))

if speed > speedlimit:
    # calculate the fine
    mphOver = speed - speedlimit
    fine = 50 + mphOver * 5

    # excessive speed
    if speed > 90:
        fine = fine + 200

    print("Your ticket is", fine)
else:
    print("Continue safe driving practices.")
```

Oct 16, 2017

Sprenkle - CSCI111

8

Using the building blocks: Nesting `if-else` statements

```
if condition :  
    if condition :  
        statements  
    else:  
        statements  
else:  
    statements
```

`if-else` statement is nested inside the `if`

Oct 16, 2017

Sprenkle - CSCI111

9

Review: Speeding Ticket Fines

- Any speed clocked over the limit results in a fine of at least \$50, plus \$5 for each mph over the limit, plus a penalty of \$200 for any speed over 90mph.
- Our program
 - Input: speed limit and the clocked speed
 - Output: either (a) that the clocked speed was under the limit or (b) the appropriate fine

What should our test cases be?
Why should those be our test cases?

Oct 16, 2017

Sprenkle - CSCI111

`speedingticket.py` 10

Testing Speeding Ticket Program

- Our test cases fell into two categories:
 - Data-related
 - Make sure we picked good numbers (clocked speed: 90, 91)
 - Control-related
 - Make sure we're hitting all the possible control-related cases, e.g., not speeding, speeding, excessive speeding

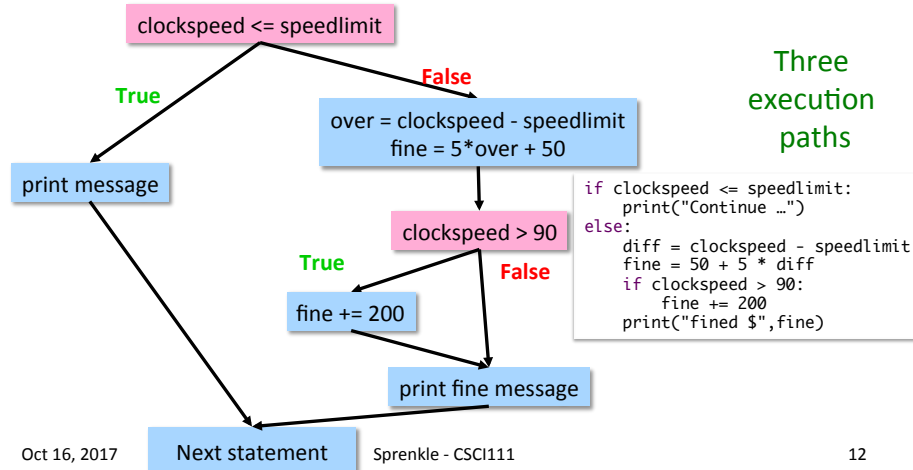
Oct 16, 2017

Sprenkle - CSCI111

speedingticket.py 11

Testing with `if` Statements

- Make sure *at least* have test cases that execute each branch in control flow diagram
 - i.e., Each execution path is "covered"



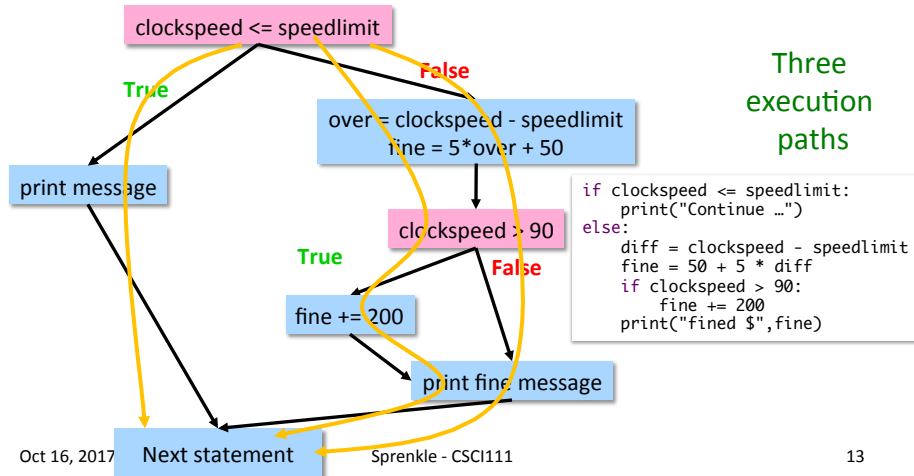
Oct 16, 2017

Sprenkle - CSCI111

12

Testing with `if` Statements

- Make sure have test cases that execute each branch in control flow diagram
 - i.e., Each execution path is “covered”



Practice: Numeric to Letter Grade

- Determine a numeric grade’s letter grade (A, B, C, D, or F)

| Numeric Grade | Letter Grade |
|----------------|--------------|
| 90 and above | A |
| 80 to below 90 | B |
| 70 to below 80 | C |
| 60 to below 70 | D |
| Below 60 | F |

Syntax of if statement: Multi-Way Decision

```
if condition :  
    <then-body1>  
elif condition :  
    <then-body2>  
elif condition :  
    <then-body3>  
...  
else:  
    <default-body>
```

keywords

English Example:

```
if it is Saturday:  
    I wake up at 10 a.m.  
elif it is Sunday:  
    I wake up at 9 a.m.  
else:  
    I wake up at 7 a.m.
```

Oct 16, 2017

Sprenkle - CSCI111

15

Using the building blocks: Nesting if-else statements

```
if condition :  
    statements  
else:  
    if condition :  
        statements  
    else :  
        statements
```

if-else statement is
nested inside the else

This structure can be rewritten as an
if-elif-else statement

Oct 16, 2017

Sprenkle - CSCI111

16

If-Else-If statements

Draw the control flow diagram

```
if x % 2 == 0 :  
    print(x, "is a multiple of 2")  
elif x % 3 == 0 :  
    print(x, "is a multiple of 3")  
else :  
    print(x, "is not a multiple of 2 or 3")
```

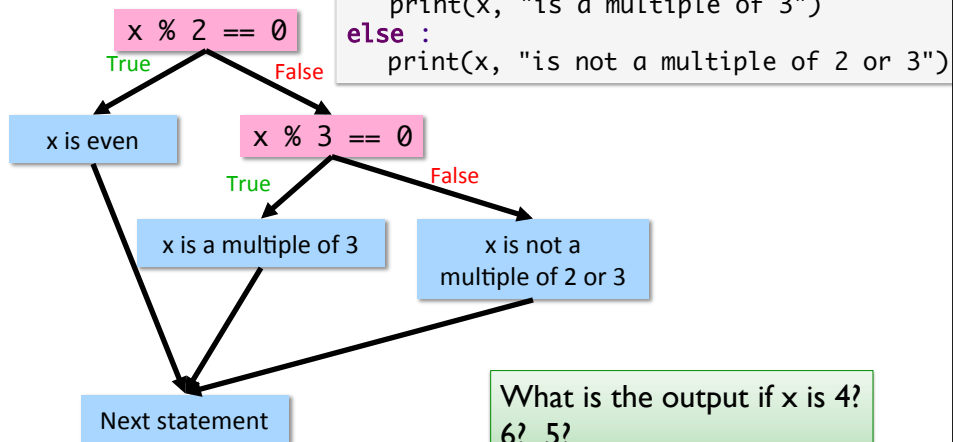
Oct 16, 2017

Sprenkle - CSCI111

17

If-Else-If statements

```
if x % 2 == 0 :  
    print(x, "is a multiple of 2")  
elif x % 3 == 0 :  
    print(x, "is a multiple of 3")  
else :  
    print(x, "is not a multiple of 2 or 3")
```



What is the output if x is 4?
6? 5?

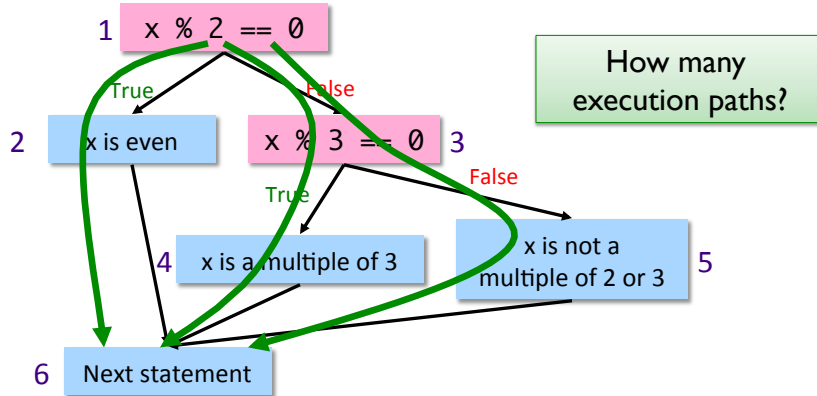
Oct 16, 2017

Sprenkle - CSCI111

18

Testing with If Statements

- Make sure have test cases that execute each branch in control flow diagram
 - i.e., Each execution path is “covered”



Oct 16, 2017

Sprenkle - CSCI111

19

Modify to use `elif`

- Determine if a numeric grade is a letter grade (A, B, C, D, or F)

| Numeric Grade | Letter Grade |
|----------------|--------------|
| 90 and above | A |
| 80 to below 90 | B |
| 70 to below 80 | C |
| 60 to below 70 | D |
| Below 60 | F |

Oct 16, 2017

Sprenkle - CSCI111

20

More Complex Conditions

- Boolean
 - Two logical values: True and False
- Combine conditions with Boolean operators
 - **and** – True only if **both** operands are True
 - **or** – True if **at least one** operand is True
 - **not** – True if the operand is not True
- English examples
 - If it is raining **and** it is cold
 - If it is Saturday **or** it is Sunday
 - If the shirt is on sale **or** the shirt is purple

Oct 16, 2017

Sprenkle - CSCI111

21

What is the output?

```
x = 2
y = 3
z = 4
```

Focus: how operations work
Not good variable names

```
b = x==2
c = not b
d = (y<4) and (z<3)
print("d=",d)
d = (y<4) or (z<3)
print("d=",d)
```

Because of precedence,
we don't need parentheses

```
d = not d
print(b, c, d)
```

Oct 16, 2017

Sprenkle - CSCI111

`eval_cond.py`

22

Truth Tables

operands

| A | B | A and B | A or B | not A | not B | not A and B | A or not B |
|---|---|---------|--------|-------|-------|-------------|------------|
| T | T | | | | | | |
| T | F | | | | | | |
| F | T | | | | | | |
| F | F | | | | | | |

Oct 16, 2017

Sprenkle - CSCI111

23

Truth Tables

operands

| A | B | A and B | A or B | not A | not B | not A and B | A or not B |
|---|---|---------|--------|-------|-------|-------------|------------|
| T | T | T | T | | | | |
| T | F | F | T | | | | |
| F | T | F | T | | | | |
| F | F | F | F | | | | |

Oct 16, 2017

Sprenkle - CSCI111

24

Truth Tables

operands

| A | B | A and B | A or B | not A | notB | not A and B | A or not B |
|---|---|---------|--------|-------|------|-------------|------------|
| T | T | T | T | F | F | | |
| T | F | F | T | F | T | | |
| F | T | F | T | T | F | | |
| F | F | F | F | T | T | | |

Oct 16, 2017

Sprenkle - CSCI111

25

Truth Tables

operands

| A | B | A and B | A or B | not A | notB | not A and B | A or not B |
|---|---|---------|--------|-------|------|-------------|------------|
| T | T | T | T | F | F | F | T |
| T | F | F | T | F | T | F | T |
| F | T | F | T | T | F | T | F |
| F | F | F | F | T | T | F | T |

Oct 16, 2017

Sprenkle - CSCI111

26

Practice: Numeric Grade Input Range

- Enforce that user must input a numeric grade between 0 and 100
 - In Python, we can't (always) write a condition like $0 \leq \text{num_grade} \leq 100$, so we need to break it into two conditions
- Write an appropriate condition for this check on the numeric grade
 - Using **and**
 - Using **or**

Oct 16, 2017

Sprenkle - CSCI111

27

Practice: Numeric Grade Input Range

- Enforce that user must input a numeric grade between 0 and 100
 - Using **and**

```
if num_grade >= 0 and num_grade <= 100:  
    computation  
else:  
    print error message
```

- Using **or**

```
if num_grade < 0 or num_grade > 100:  
    print error message  
else:  
    computation
```

Oct 16, 2017

Sprenkle - CSCI111

28

Short-circuit Evaluation

- Don't necessarily need to evaluate all expressions in a compound expression
- A **and** B
 - If A is **False**, compound expression is **False**
- A **or** B
 - If A is **True**, compound expression is **True**
- No need to evaluate B
 - Put more important/limiting expression first
 - Example:

```
if count != 0 and sum/count > 10:  
do something
```

Oct 16, 2017

Sprenkle - CSCI111

29

Exam 1

- Overall, went very well
- Review mistakes and how to correct
- Don't rest on your laurels

| | Sections | | | Total |
|---------|----------|-------|-------|-------|
| | A | B | C | |
| Average | 90.50 | 75.93 | 92.01 | 92.48 |
| Median | 93.02 | 77.78 | 95.35 | 92.50 |

Oct 16, 2017

Sprenkle - CSCI111

30

Looking Ahead

- Pre lab 5 due tomorrow, before lab
- Lab 5 tomorrow
- BI – posted later