

Objectives

- More Conditionals
- Boolean Operators
- Creating modules

1

Review

- How can we make Python code execute only under certain circumstances?
- How do we say “otherwise” in Python?
- What are relational operators?
 - Provide examples

2

Review: 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

Feb 14, 2022

Sprenkle - CSCI111

3

3

Review: Two-Way Decision

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

keywords

“then” Body

“else” Body

English Example:

if it is Saturday or Sunday :
 I wake up at 9 a.m.

else :
 I wake up at 7 a.m.

Feb 14, 2022

Sprenkle - CSCI111

4

4

Review: Relational Operators

- Syntax: <expr> <relational_operator> <expr>
- Evaluates to either True or False
 - Boolean type

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

Low precedence
After arithmetic operators

Feb 14, 2022

Sprenkle - CSCI111

Use Python interpreter

5

5

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?

Feb 14, 2022

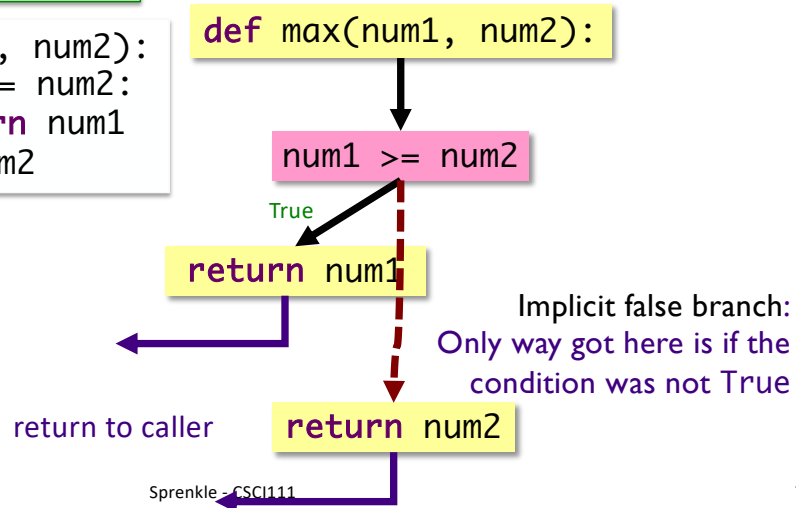
6

6

Review: Flow of Control: Using `return`

Is this implementation of the function correct?

```
def max(num1, num2):  
    if num1 >= num2:  
        return num1  
    return num2
```



Feb 14, 2022

Sprenkle - CSCI111

7

7

Practice: 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 function
 - Input: speed limit and the clocked speed
 - Output: the appropriate fine
 - What should the appropriate fine be if the user is not speeding?

Feb 14, 2022

Sprenkle - CSCI111

[speedingticket.py](#)

8

8

Test-Driven Development (TDD)

- Create test cases first
- Idea: Focus on the outcomes first
- Helps you think about the problem without thinking about the code itself

Feb 14, 2022

Sprenkle - CSCI111

9

9

Testing Speeding Ticket Program

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

Feb 14, 2022

Sprenkle - CSCI111

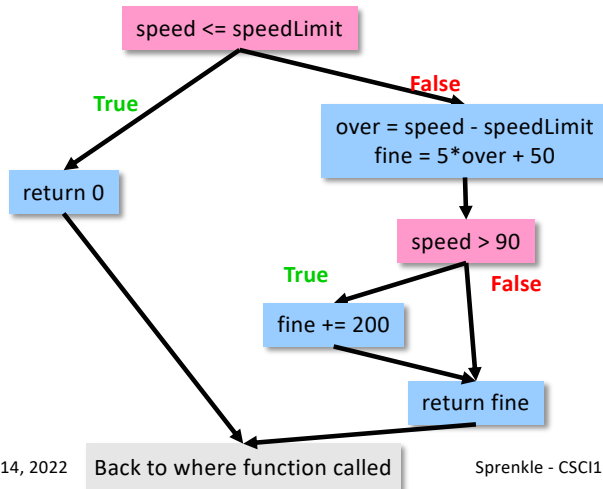
speedingticket.py

10

10

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”



Three execution paths

```
if speed <= speedLimit:
    return 0
else:
    diff = speed - speedLimit
    fine = 50 + 5 * diff
    if speed > 90:
        fine += 200
    return fine
```

Feb 14, 2022

Back to where function called

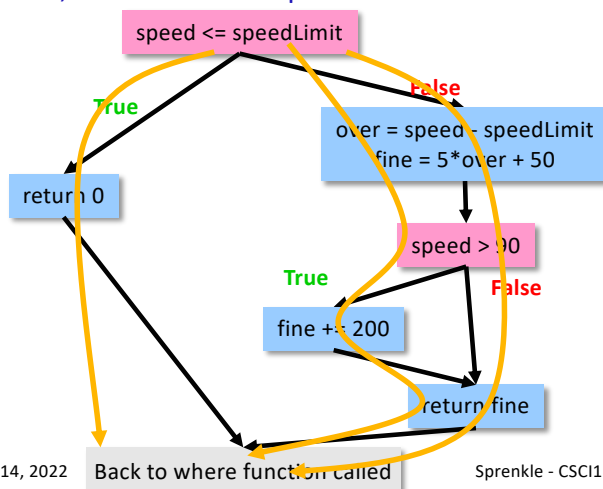
Sprenkle - CSCI111

11

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”



Three execution paths

```
if speed <= speedLimit:
    return 0
else:
    diff = speed - speedLimit
    fine = 50 + 5 * diff
    if speed > 90:
        fine += 200
    return fine
```

Feb 14, 2022

Back to where function called

Sprenkle - CSCI111

12

12

Practice: Speeding Ticket Fines

- Any speed over the limit is at least \$50, plus a penalty of \$5 for every mile per hour over the limit.

```
def main():  
    print("This program ...")  
  
    clockedSpeed = eval(input("Enter your speed: "))  
    speedLimit = eval(input("Enter the speed limit: "))  
  
    # your code here  
  
def calculateFine(limit, speed):  
    ...
```

- Our program

- Input: speed limit and the clocked speed
- Output: appropriate output to the user, *based on their speeding/fine*

speedingticket.py

Feb 14, 2022

Sprenkle - CSCI111

13

13

Using the building blocks: Nesting if-else statements

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

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

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

Feb 14, 2022

Sprenkle - CSCI111

14

14

Practice: Numeric to Letter Grade

- Write a program to 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

```
numericGrade = float(input("Numeric grade: "))  
#...  
print("Your grade is", letterGrade)
```

Feb 14, 2022 grade.py

15

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.
```

Feb 14, 2022

Sprengle - CSCI111

16

16

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

Feb 14, 2022

Sprenkle - CSCI111

17

17

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")
```

Feb 14, 2022

Sprenkle - CSCI111

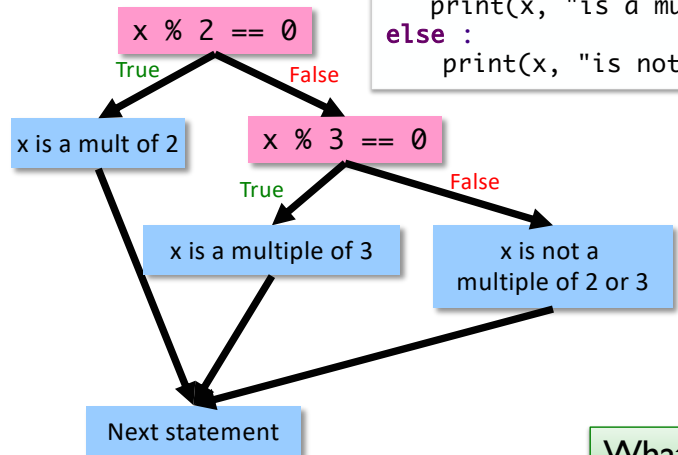
18

18

If-Else-If statements

```

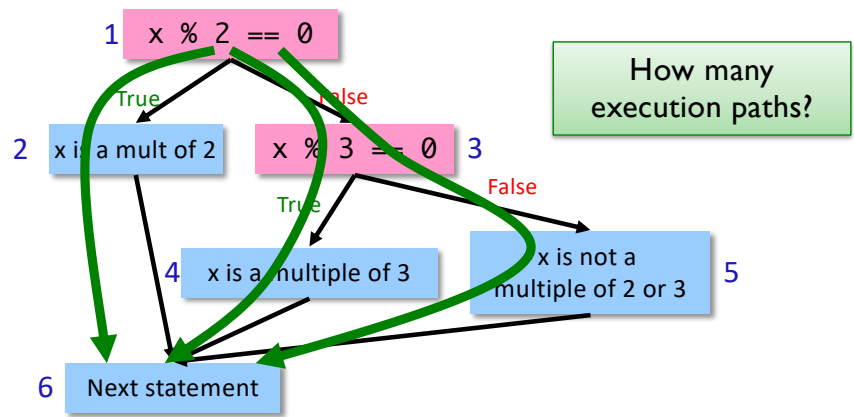
if x % 2 == 0 :
    print(x, "is a multiple of 2")
elif x % 3 == 0 :
    print(x, "is a multiple of 3")
else :
    print(x, "is not a multiple of 2 or 3")
    
```



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

Testing with If Statements

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



How many execution paths?

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

Feb 14, 2022

Sprenkle - CSCI111

21

21

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

Feb 14, 2022

Sprenkle - CSCI111

22

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						

Feb 14, 2022

Sprenkle - CSCI111

23

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				

Feb 14, 2022

Sprenkle - CSCI111

24

24

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	F	F		
T	F	F	T	F	T		
F	T	F	T	T	F		
F	F	F	F	T	T		

Feb 14, 2022

Sprenkle - CSCI111

25

25

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	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

Feb 14, 2022

Sprenkle - CSCI111

26

26

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)
```

Feb 14, 2022

Sprenkle - CSCI111

`eval_cond.py`

27

27

Looking Ahead

- Pre lab 5 due tomorrow, before lab
 - Don't miss the last section from Functions chapter
- Lab 5 tomorrow
- BI: self-driving cars

Feb 14, 2022

Sprenkle - CSCI111

28

28