

Objectives

- More on conditionals
- Indefinite Loops

Jan 30, 2007

Sprengle - CS111

1

Review: Relational Operators

- Syntax:
 - <expression> <relational_operator><expression>

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

Low precedence

Jan 30, 2007

Sprengle - CS111

2

Enhanced Lottery Game

- Check if user's pick matches the number you generated
- When using strings, needed the user's input to be in the same format
- The debate:
 - User's input as a string or as a number --> create the lottery number a string or number

Jan 30, 2007

Sprengle - CS111

pick4winner2.py

3

Enhanced Lottery Game

- Check if user's pick matches the number you generated
- When using strings, needed the user's input to be in the same format
- The debate:
 - User's input as a string or as a number --> create the lottery number a string or number
- *Almost* works with numbers
 - Demonstrate two "near" solutions
 - Breaks down when first number is 0

Jan 30, 2007

Sprengle - CS111

pick4winner2.py

4

Modify: Check for Other Bad Input

```
print "This program determines your birth year"
print "given your age and current year"
print
age = input("Enter your age >> ")
if age > 110:
    print "Don't be ridiculous, you can't be that old."
else:
    currentYear = input("Enter the current year >> ")
    birthyear = currentYear - age
    print
    print "You were either born in", birthyear, "or", birthyear-1
```

Jan 30, 2007

Sprengle - CS111

5

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:

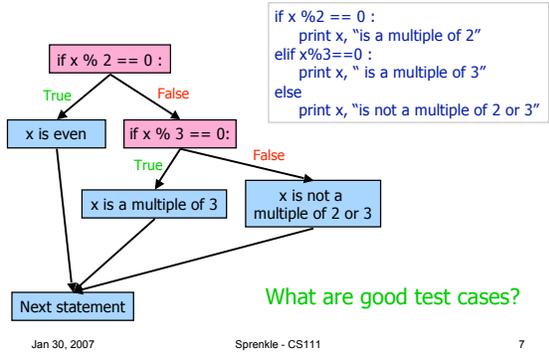
```
if it is Saturday:
    I wake up at 10 a.m.
elif it is Sunday:
    I wake up at 9 a.m.
else :
    I wake up at 7 a.m.
```

Jan 30, 2007

Sprengle - CS111

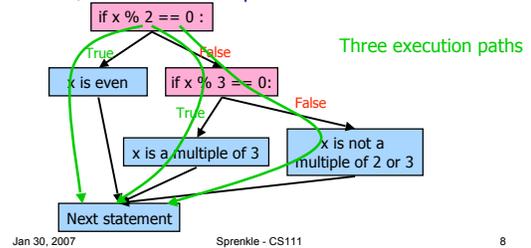
6

If-Else-If statements



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 the letter grade (A, B, C, D, or F) for a numeric grade

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

Jan 30, 2007 Sprenkle - CS111 9

Using the building blocks: nesting if-else statements

```

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

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

Jan 30, 2007 Sprenkle - CS111 10

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

Jan 30, 2007 Sprenkle - CS111 11

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

Convert the Code to if-elif-else

```

clockspeed = input("Enter the clocked speed: ")
speedlimit = input("Enter the speed limit: ")

if clockspeed <= speedlimit:
    print "Continue safe driving practices"
else:
    diff = clockspeed - speedlimit
    fine = 50 + 5 * diff
    if clockspeed > 90:
        fine += 200
    print "Slow down! You've been fined $" + str(fine) + "."
    
```

Jan 30, 2007 Sprenkle - CS111 12

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

Jan 30, 2007

Sprenkle - CS111

13

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						

Jan 30, 2007

Sprenkle - CS111

14

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

Jan 30, 2007

Sprenkle - CS111

15

What is the output?

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

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

```
print b, c, d
```

Jan 30, 2007

Sprenkle - CS111

eval_cond.py

16

Practice: Numeric Grade Input Range

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

Jan 30, 2007

Sprenkle - CS111

17

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
```

Jan 30, 2007

Sprenkle - CS111

18

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`

Jan 30, 2007

Sprengle - CS111

19

Indefinite Loops

- **for** loops are **definite** loops
 - Execute a *fixed* number of times
- Indefinite loops: keeps iterating until certain conditions are met
 - Depending on condition, no guarantee in advance of how many times the loop body will be executed

Jan 30, 2007

Sprengle - CS111

20

While Loop Syntax

```

while condition :
    statement1
    statement2
    ...
    statementn
    
```

keyword

body of while loop

- Like a looped **if** statement
 - Execute statements **only** when condition is true

Jan 30, 2007

Sprengle - CS111

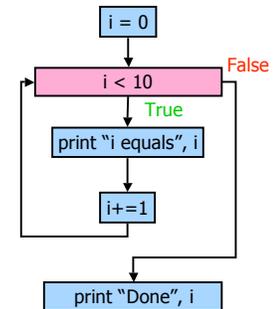
21

While Loop

```

i = 0
while i < 10 :
    print "i equals ", i
    i += 1
print "Done", i
    
```

- Questions:
- How many times will "i" get printed out?
 - How many times is the condition evaluated?
 - What is the value of i after the loop?



Jan 30, 2007

Sprengle - CS111

while.py

22

While Loop

```

i = 0
while i < 10 :
    print "i equals ", i
    i += 1
print "Done", i
    
```

Initialize i before using in condition

- Questions:
- How many times will "i" get printed out?
 - How many times is the condition evaluated?
 - What is the value of i after the loop?

Jan 30, 2007

Sprengle - CS111

while.py

23

While vs. For Loops

- Any **for** loop can be translated into a **while** loop
 - **Not vice versa**
- **while** loops are more **powerful** than **for** loops

Jan 30, 2007

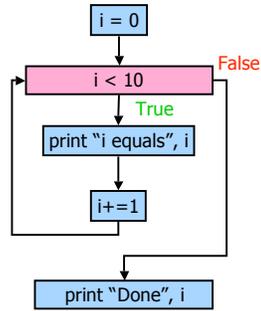
Sprengle - CS111

24

Convert to a **for** loop

We can convert this while loop into a for loop because it executes a **fixed** number of times.

```
i = 0
while i < 10 :
    print "i equals ", i
    i+=1
print "Done", i
```



What are the main differences?

Jan 30, 2007

Sprengle - CS111 [whilevsfor.py](#)

25

Comparing **while** and **for**

- What are the main differences between these loops?
- What are the advantages and disadvantages of each?

```
i = 0
while i < 10 :
    print "i equals ", i
    i+=1
print "Done", i
```

```
for i in xrange(10):
    print "i equals", i
print "Done", i
```

Jan 30, 2007

Sprengle - CS111 [whilevsfor.py](#)

26

What Will This Loop Do?

```
count = 1
while count > 0:
    print count
    count += 1
```

Jan 30, 2007

Sprengle - CS111

27

Infinite Loop

- Condition will never be False so keeps executing
- To stop an executing program in Linux use
➤ **Control-C**

Jan 30, 2007

Sprengle - CS111

28

For Friday

- Broader Issue: DARPA Urban Challenge

Jan 30, 2007

Sprengle - CS111

29