Flow of Control

Repetition
Statements can be repeated some fixed number of time, or else can be repeated until some event signals they should not be repeated any more.

Overview/Questions

– How do we control the flow of execution within our programs?
– Review: the definite loop.
– Is there an indefinite loop?
– When bad things happen to good programs...

Review: Definite Loop

The definite loop is used to iterate values in a sequence, and has the general form:

```
for <var> in <sequence>:
    <body>
```

The variable `<var>` after the keyword `for` is called the loop index. It will assume each successive value in `<sequence>`, and will perform the `<body>` statements for each value.
How many times to loop?

The definite loop must operate on a sequence. That is, it needs to know in advance how many times to repeat.

- What if we don’t know how many repetitions we’ll need?

Example: averaging quiz grades

- Definite loop version

```python
n = input('How many numbers to average?: ')
# create an accumulator variable:
total = 0
# repeat n times:
for i in range(n):
    # collect input, add to accumulator 'total'
    num = input('Enter the next input: ')
    print('Adding value', num)
    total = total + num
# calculate the average
print('The total is:', total)
average = float(total) / n
print('The average is', average)
```

Review: Definite Loop

Example: averaging quiz grades

- definite loop version

The Indefinite Loop

The indefinite loop is controlled by a conditional expression.

```
while statement
A structure which evaluates a logical expression, and repeats a controlled block of statements.
```

Patterns for Indefinite Loops

Interactive Loop
Prompts user whether or not to continue.

- set moredata to “yes”
- while moredata is “yes”
  - get next data item
  - process data item
  - ask user if there is moredata
The Indefinite Loop

Example: averaging quiz grades
- indefinite loop version, interactive pattern.

```
# create an accumulator variable:
total = 0
# keep a count of how many entries we've got
count = 0
# prompt user about continuing:
moredata = raw_input('Got more data? (yes/no): ')
while moredata == 'yes':
    num = input('Enter the next input: ')
    print 'Adding value', num
    total = total + num
    count += 1 # add one for each entry received
    # prompt user about continuing:
    moredata = raw_input('Got more data? (yes/no): ')
print 'The total is', total
average = float(total) / count
print 'The average is', average
```

Patterns for Indefinite Loops

Sentinel Loop
Checks input for a special value to determine whether or not to continue.

```
get first data item
while data item is not the sentinel
    process data item
    get next data item
```

The Indefinite Loop

Example: 99 bottles of beer song
- indefinite loop version

```
bottles = input('What's the biggest number you know? ') 
print # blank line
# continue until all the bottles are down
while bottles >= 0:
    print bottles, 'bottles of beer on the wall.'
    print bottles, 'bottles of beer.'
    bottles = bottles - 1 # adjust the counter downward
    print bottles, 'bottles of beer on the wall.'
# wait for user to hit enter:
input = raw_input('>')
print # blank line
print 'Thank god that's over!
```
Indefinite Loop Problems

Infinite Loop
A loop with no end – no way out!
Generally, this is a very bad problem.

Example:
an argument with Aaron’s friend Johannes

Busy Loop
An infinite loop, which never pauses (e.g. for input), and thus consumes all of the computer’s resources.
This is a terrible, awful, no-good, very bad problem.

Example:
a counting loop with a faulty condition.
Application: Validating User Input

Ensure that user enters a positive number:

```python
age = input("Enter your age: ")
while age <= 0:
    age = input("Enter your age (as a positive integer): ")
print "Your age is", age, "years."
```

Summary

- The definite loop must operate on a sequence. That is, it needs to know in advance how many times to repeat.
- Indefinite loops are controlled by a condition, and continuation or stopping can be controlled interactively.
- Infinite loops and busy loops are major risks of using indefinite loops.

Student To Dos

- HW 03 due TODAY!
- HW 04 will be on Python while
- QUIZ 2 on Wednesday!