

CS150 - Exceptions & Error Handling

Note Title

2/15/2012

Announcements

- Review for midterm - next Friday
Midterm on Monday the 27th
- HW due Tuesday (not Monday)

Practice 5.31

Write a function `yesOrNo(prompt)` that asks the user a question (specified in the string `prompt`) and demands a response of 'yes' or 'no'.

If wrong input is received, ask user again for yes or no answer. (repeat until you get one.)
Return true if user says yes,
* false if user says 'no'.

More on checking input:

Want to have user enter a number
between 1 & 10:

```
nbr = 0  
while not nbr in range(1:11):  
    nbr = int(raw_input("Enter a number between 1 and 10:"))
```

OR:

```
nbr = 0  
while not 1 <= nbr <= 10 :  
    nbr = int( ' ' ' ' ' ' )
```

Catching exceptions

ValueError

In previous example, if the user enters a string, the code halts with an error.

This tells us what is happening, but in software that production quality, ending abruptly with an obscure error message is unacceptable.

Better:

Print appropriate message & retry.

Catching exceptions : try statements

try :

code which could cause error

except errorClass :

code to gracefully recover

How it works: runs code in try.
IF error happens, goes to except code.

Example

```
number = 0
while not 1 <= number <= 10:
    try:
        number = int(input('Enter ... 1 to 10:'))
        if not 1 <= number <= 10:
            print 'Your number must be from 1 to 10'
    except ValueError:
        print 'That is not a valid integer.'
```

Type checking

```
def scaleData(data, factor):  
    for i in range(len(data)):  
        data[i] = factor * data[i]
```

list

float or int

What could get wrong?

```
if not isinstance(data, list):  
    raise TypeError(  
if not isinstance(factor, (int, float)):
```

Practice Problem

HW2 revisited:

Go back and make functions based on your code from problems 1 and 5.

Add type checking - be sure to raise or catch exceptions as needed.