CS150 - Exceptions & Error Handling

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 and 10:

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

OR:

```
nbr = 0
while not 1 <= nbr <= 10:
    nbr = int(input('Enter a number between 1 and 10: '))
```
Catching exceptions

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 produces quality, ending abruptly with an obscure error message is unacceptable.

Better:

Print appropriate message and retry.
Catching exceptions: try statements

try:
    [code which could cause error
    except error class:
        [code to gracefully recover

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

```python
number = 0
while not 1 <= number <= 10:
    try:
        number = int(input('Enter a number from 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

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

What could go wrong?
```

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

HWA 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.