CS 160 CS Orientation

Strings and Bad Input



Odds and Ends...

- Demo Assignment #4
- Assignment #5 posted
- Cool code visualizer:
- <u>http://pythontutor.com/visualize.html#mode=</u>
 <u>edit</u>



Revisit Square Root Finder...

Design a Python program that takes a **positive whole number** *n* as input and **outputs the square root of n** using the Babylonian algorithm. The Babylonian algorithm computes the square root of a positive number, *n*, as follows:

- 1. Make a guess at the answer (you can pick n/2 as your initial guess).
- 2. Compute r = n / guess
- 3. Set guess = (guess + r) / 2
- 4. Go back to step 2 for as many iterations as necessary. The more steps 2 and 3 are repeated, the closer guess will become to the square root of n.

SI Oregon State University

In-class Exercise: check loop understanding

- Get into groups of 4-5.
- Explain what each of line in this code does and tell me what the output is:

```
y=0;
for x in range(10):
    if(x%2):
        y=y+1;
print(y);
```



What happens when you enter bad data?

• Where/when does the error occur?



Strings

- Create a string my_string="hello";
- Access a character my_string[0] #gives you first character
- Length len(my string)



Exercise

- Write an algorithm to determine if input is bad without using exceptions, i.e. it would work in any language!!! ^(C)
 - How will you make sure you get a good positive integer from the user?
 - How will you make sure you get a good floating point number from the user?

