CS 160
CS Orientation

Practice Loops and Functions
Odds and Ends...

• Assignment #7 posted!!!
• Assignment #6 demoed this week!!!

• Questions About Anything???
In-class Exercise #4

Design a Python function called my_sqrt() that takes a positive whole number \( n \) as input and returns 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 \( r = n / \text{guess} \)
3. Set \( \text{guess} = (\text{guess} + r) / 2 \)
4. Go back to step 2 for as many iterations as necessary. The more steps 2 and 3 are repeated, the closer \( \text{guess} \) will become to the square root of \( n \).
5. Compare your square root function with the math.sqrt() result.
Stop when threshold reached, but what if we didn’t know the exact val...

```python
# In-class exercise 4

def get_int():
    while True:
        bad_input = True;
        num = input("Enter positive integer: ");
        for x in range(len(num)):
            if num[x] < '0' or num[x] > '9':
                print("Bad input.");
                bad_input = False;
                break;
        if bad_input == True:
            return int(num);

def my_sqrt():
    num = get_int();
    guess = num / 2;
    while not (num - (guess * guess) > -0.1 and num - (guess * guess) < 0.1):
        r = num / guess;
        guess = (guess + r) / 2;
    return guess;

def main():
    answer = my_sqrt();
    print("Answer is: " + str(answer));
main();
```
Stop when threshold reached, but what if we didn’t have sqrt() to use…

```python
from math import sqrt

def is_pos_int(x):
    for i in x:
        if i < '0' or i > '9':
            return False;
    return True;

def calculation(n, guess):
    r = n / guess;
    guess = (guess + r) / 2;
    return guess;

def main():
    number = input("What number would you like to take the square root of?: ");
    if is_pos_int(number) == False:
        print("That's not a valid input.");
        main();
    a = 1.0;
    number = float(number);
    while abs(a - sqrt(number)) > .001:
        a = calculation(number, a);
    print("The square root of " + str(number) + " is " + str(a));
main();
```

"sqrt.py" 24L, 538C
Stop when max iterations reached...

```python
def main():
    n=input("Enter a positive whole number: ");
    if len(n)==0:
        print("Error")
        exit();
    for i in range(len(n)):
        if n[i]>'9' or n[i]<'0':
            print("Error")
            exit();
    n=int(n)
    z=int(0);
    g = float(n / 2)
    while z < 50:
        r = float(n / g)
        g = (g + r) / 2
        z = z + 1
    print("The square root of " + str(n) + " is " + str(g) + ".")
main();
```

"babylonianroot.py" 18L, 354C
Try making a new_guess and old_guess to test for threshold reached...

• Modify your solution to try this...