CS 160
CS Orientation

Designing Functions
Odds and Ends…

• Assignment #7 questions?
• Use your TA office hours!!!
• Assignment #6 demoed this week!!!
Revisit Exercise #6

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 guess will become to the square root of \( n \).
5. Compare your square root function with the math.sqrt() result.
Demo...

```python
1 def my_sqrt(n): # n = num
2     guess = n / 2
3     # Also, you could ask user for iterations or stop when a threshold is reached, such as absolute value of new_guess-old_guess < .0000001
4     for i in range(1000):
5         r = n/guess
6         guess = (guess + r)/2
7     return guess

9 # main function of where to start our program
10 def main():
11     num=int(input("Enter positive num: "));
12     my_sqrt(num);
13
14 main(); # start by calling main function
```

-- INSERT --
Revisit function design...

- For all the employees in our company, calculate their gross pay based on their hours and pay rate.

- Create a function to get number of employees.  
  - Make sure this function only returns a valid integer.

- Create function called gross_pay that takes the number of employees and calculates gross pay for each.

- Create a main function where the program begins.
Designing Functions...

description: Takes a string and returns true if all digits 0-9, otherwise false

Parameters:
1st: String of chars

Precondition: has 1 > chars

Returns: true if Q, false, otherwise
def is_pos_int(s):
    for i in range(len(s)):
        if s[i]<'0' or s[i]>'9':
            return False;

    return True;

# while(1):
#    st=input("Enter good positive int: ");
#    print(is_pos_int(st));

-- INSERT --
import math
import my_math

def my_sqrt(n):  # n=number
    guess = n / 2
    # Also, you could ask user for iterations or stop when a threshold is reached, such as absolute value of new_guess-old_guess < .0000001
    for i in range(1000):
        r = n/guess
        guess = (guess + r)/2
    return guess

# main function of where to start our program
def main():

    num=input("Enter positive num: ");
    while(not my_math.is_pos_int(num)):
        print("You idiot, I said a good positive int!!!");
        num=input("Enter positive num: ");

    print("my sqrt: "+str(my_sqrt(int(num))));
    print("pre defined sqrt: "+str(math.sqrt(int(num))));

main();  # start by calling main function
Turtle Example...

```python
import turtle

window=turtle.Screen();
my_turt=turtle.Turtle();
my_turt2=turtle.Turtle();

my_turt.shape("turtle");
my_turt.color("green");

my_turt.setposition(-30, -30);

window.mainloop();
```