CS 160
CS Orientation

Creating Functions...
As a class, let’s define functions...

• 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.

• What if you want to 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.
Define a `get_num_emp` function

```python
def get_num_emp():
    num_emp = input("enter num employees: ");
    error = False;  # we assume we don't have an error
    # all characters in string must be 0-9 digits to be positive int
    for x in range(len(num_emp)):
        if (num_emp[x] < '0' or num_emp[x] > '9'):
            print("You idiot! I said integer number!!");
            error = True;  # We found an error!
            break;  # quit looking at characters when one is found to be invalid
    print(x);  # what is the value of x after the loop
    print(len(num_emp));
    #if(not num_emp.isnumeric()):
    #    print("You idiot! I said integer number!!");
    #else:
    if(error == False):  # if there wasn't an error, then valid
        return int(num_emp);
    else:  # what do you want to return when bad? Could be -1
        return 0;
```

--- INSERT ---

OSU Oregon State University
return vs. break

def get_num_emp():
    num_emp = input("enter num employees: ");
    # all characters in string must be 0-9 digits to be positive int
    for x in range(len(num_emp)):
        if x < '0' or x > '9':
            print("You idiot! I said integer number!!");
            # break; # quit looking at characters when one is found to be invalid
            return -1; # leave the loop and function if it isn't good!

    return int(num_emp);

def main():
    num_emp = get_num_emp();
    for x in range(num_emp):
        hours = float(input("Employee #" + str(x+1) + " enter hour: "));
        pay = float(input("enter pay: "));
        gross = hours * pay;
        print(gross);

main();

OSU Oregon State University
As a class, let’s define functions...

• 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.

• What if you want to 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...

main

\( n = \text{get\_num\_emp()} \)

\( \text{calc\_gross\_pay}(n) \)

get\_num\_emp

Description:
set a valid positive integer value for # of employees

Preconditions:
what holds true about mem/args coming to function

None

Postconditions:
- after function is called
- holds true about mem/args

Returns:
valid positive integer

is valid int()
def is_valid_int(num):
    # all characters in string must be 0-9 digits to be positive int
    for x in range(len(num)):
        if(num[x]<'0' or num[x]>'9'):
            print("You idiot! I said integer number!!");
            # break; quit looking at characters when one is found to be invalid
            return False;
    return True;

def get_num_emp():
    num_emp=input("enter num employees: ");
    while(not is_valid_int(num_emp)):
        num_emp=input("enter num employees: ");
    return int(num_emp);

def main():
    num_emp=get_num_emp();
    for x in range(num_emp):
        hours=float(input("Employee #"+str(x+1)+" enter hour: "));
        pay=float(input("enter pay: "));
        gross=hours*pay;
        print(gross);

main();
Can we change information in functions?

```python
def is_valid_int(num):
    # all characters in string must be 0-9 digits to be positive int
    for x in range(len(num)):
        if (num[x]<'0' or num[x]>'9'):
            print("You idiot! I said integer number!!");
            #break; #quit looking at characters when one is found to be invalid
            return False;
    return True;

def get_num_emp():
    num_emp=input("enter num employees: ");
    while(not is_valid_int(num_emp)):
        num_emp=input("enter num employees: ");
    return int(num_emp);

def fun(n):
    n=5; #cannot change an immutable type inside function

def main():
    num_emp=get_num_emp();
    print(num_emp);
    fun(num_emp);
    print(num_emp);
    for x in range(num_emp):
        hours=float(input("Employee #"+str(x+1)+" enter hour: "));
        pay=float(input("enter pay: "));
        gross=hours*pay;
        print(gross);

main();
```