CS 160
CS Orientation
Programming Structure & Functions
Chap. 4 & 5
What is pseudocode/design?

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

  ```
prompt for #employees
read #employees + assign to vari
for all employees
  prompt/read hours
  prompt/read pay
  print/calculate of hours * pay
  ```
Python Demo

```python
employees = int(input("Enter num of employees: "))

for x in range(employees):
    hours = float(input("Enter Employee #"+str(x+1)+"'s hours: "))
    pay = float(input("Enter Employee #"+str(x+1)+"'s pay: "))
    print("Employee #"+str(x+1)+" has gross pay "+str(hours*pay))
```
What if the problem changed?

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

• Required functions:
  get_num_employees()
  get_employee_info()
  calculate_gross_pay()
  print_gross_pay()
  main()
# Description: prompt and receive number of employees
# Parameters: none
# Returns: number of employees
# Pre-conditions: none
# Post-conditions: none

# Define get_num_employees
    #prompt for number of employees
    #read and assign to variable

    #return variable

# Description: prompt and receive pay rate and hour info for the employees
# Parameters: list of employee pay rate, list of employee hours, number of employees
# Returns: none
# Pre-conditions: there would be two empty lists for pay rate and hours
# Post-conditions: both lists would no longer be empty

# Define get_employee_info
    #for all employees
    #prompt user for pay rate
    #read the pay rate for the employee
    #prompt user for hours
    #read the hours for the employee
Implementing Design Demo...

```python
# Description: prompt and receive number of employees
# Parameters: none
# Returns: number of employees
# Pre-conditions: none
# Post-conditions: none

# Define get_num_employees
def get_num_employees():
    # prompt for number of employees
    # read and assign to variable
    n = int(input("Enter number of employees: "))

    # return variable
    return n
```
Implement more design...

```python
# Description: prompt and receive pay rate and hour info for
# the employees
# Parameters: list of employee pay rate, list of employee of
# hours, number of employees
# Returns: none
# Pre-conditions: there would be two empty lists for pay rate
# and hours
# Post-conditions: both lists would no longer be empty

# Define get_employee_info
def get_employee_info(p, h):
    # for all employees
    for x in range(len(p)):
        # prompt user for pay rate
        # read the pay rate for the employee
        p[x] = float(input("Enter pay for Employee #"+str(x+1)+": "))
        # prompt user for hours
        # read the hours for the employee
        h[x] = float(input("Enter hours for Employee #"+str(x+1)+": "))
```

Test functions with main()...

```python
# Main function calls and tests other functions. Needs to be below the
# definitions of the functions that it calls

def main():
    num_emp = get_num_employees()

    pay = [0] * num_emp
    hours = [0] * num_emp
    print(pay)
    print(hours)

    get_employee_info(pay, hours)
    print(pay)
    print(hours)

main()  # Make sure to call your main function to start the program:)
```