

## Assignments

2.2 Write a program that uses input to prompt a user for their name and then welcomes them.

Note that input will pop up a dialog box. Enter Sarah in the pop-up box when you are prompted so your output will match the desired output.

# Desired Output

# Hello Sarah

```
name = input("What's your name?")
print "Hello "+name
```

2.3 Write a program to prompt the user for hours and rate per hour using input to compute gross pay.

Use 35 hours and a rate of 2.75 per hour to test the program (the pay should be 96.25). You should use input to read a string and float() to convert the string to a number. Do not worry about error checking or bad user data.

# Desired Output

# 96.25

```
hrs = input("Enter Hours:")
rate = input("Enter rate:")
result = float(rate)*float(hrs)
print result
```

3.1 Write a program to prompt the user for hours and rate per hour using raw\_input to compute gross pay.

Pay the hourly rate for the hours up to 40 and 1.5 times the hourly rate for all hours worked above 40 hours.

Use 45 hours and a rate of 10.50 per hour to test the program (the pay should be 498.75).

You should use input to read a string and float() to convert the string to a number. Do not worry about error checking the user input - assume the user types numbers properly.

# Input: hrs=45, rate=10.5

# Desired Output: 498.75

```
hrs = input("Enter Hours:")
h = float(hrs)
```

```

rate = input("Enter Rate:")
r = float(rate)

if h>40:
    result = 40*r+(h-40)*r*1.5
else:
    result = h*r

print result

```

3.3 Write a program to prompt for a score between 0.0 and 1.0.  
If the score is out of range, print an error.

If the score is between 0.0 and 1.0, print a grade using the following table:

Score Grade

>= 0.9 A

>= 0.8 B

>= 0.7 C

>= 0.6 D

< 0.6 F

If the user enters a value out of range, print a suitable error message and exit. For the test, enter a score of 0.85.

# Input: 0.85

# Desired Output: B

```

score = raw_input("what's the score:")
s = float(score)

```

```

if s<0.0 or s>1.0:

```

```

    print "error"

```

```

elif s<0.6:

```

```

    print "F"

```

```

elif s<0.7:

```

```

    print "D"

```

```

elif s<0.8:

```

```

    print "C"

```

```

elif s<0.9:

```

```

    print "B"

```

```

else:

```

```

    print "A"

```

4.6 Write a program to prompt the user for hours and rate per hour using `raw_input` to compute gross pay.

Award time-and-a-half for the hourly rate for all hours worked above 40 hours.

Put the logic to do the computation of time-and-a-half in a function called `compute_pay()` and use the function to do the computation.

The function should return a value.

Use 45 hours and a rate of 10.50 per hour to test the program (the pay should be 498.75).

You should use `input` to read a string and `float()` to convert the string to a number. Do not worry about error checking the user input unless you want to - you can assume the user types numbers properly.

# Input: hrs=45, rate=10.5

# Output: 498.75

```
def compute_pay(h,r):  
    if h<=40:  
        return h*r  
    else:  
        return 40*r+(h-40)*r*1.5
```

```
hrs = input("Enter Hours:")  
rate = input("Enter Rate:")
```

```
hr = float(hrs)  
ra = float(rate)
```

```
p = compute_pay(hr,ra)
```

```
print p
```

5.2 Write a program that repeatedly prompts a user for integer numbers until the user enters 'done'.

Once 'done' is entered, print out the largest and smallest of the numbers.

If the user enters anything other than a valid number catch it with a `try/except` and put out an appropriate message and ignore the number.

Enter the numbers from the book for problem 5.1 and Match the desired output as shown.

```
# Input: 5,6,hehe,4,7
# Desired Output:
#     Invalid input
#     Maximum is 7
#     Minimum is 4
```

```
largest = None
smallest = None
```

```
while True:
    num = input("Enter a number: ")
    if num == "done" : break

    try:
        n=int(num)

    except:
        print "Invalid input"

    if n>largest:
        largest=n
    if smallest is None or n<smallest:
        smallest=n
    #elif n<smallest:
    #smallest=n

print "Maximum is", largest
print "Minimum is", smallest
```

# Programming for Everybody (Getting Started with Python)



Completed by **Shao Yu Liu**

April 5, 2020

2-4 hours/week

Grade Achieved: 100.00%

Shao Yu Liu's account is verified. Coursera certifies their successful completion of [Programming for Everybody \(Getting Started with Python\)](#)



**Programming for Everybody (Getting Started with Python)**

University of Michigan

★★★★★ 4.8 (135,693 ratings) | 1.5M Students Enrolled

