Lab 2 – Conditions and Tuples

Note:

- Download the Jupyter notebook “Week 3 In-Class Exercises Starting Code (v1.0).ipynb” (which contains the starting code for the exercises below) from eLearn.

Q1: Code Tracing

Take a look at the following code. What do you think will be the output of the code?

a)

```
a = 20
b = 20
if a >= b:
    print("a >= b")
if a <= b:
    print("a <= b")
```

b)

```
a = 30
b = 30
if a >= b:
    print("a >= b")
elif a <= b:
    print("a <= b")
```

c)

```
c = "ACCT649"
d = "acct649"
e = "ACCT" + "649"

print(c == d)
print(c == e)
```

d)

```
def test_if_else(condition1, condition2):
    if (condition1):
        print("Great!")
        return True
    elif(condition2):
        print("Good!")
        return True
    else:
        print("Okay")
        return False

result = test_if_else(4 % 2 != 0, 3 // 2 == 1)
print(result)
```

Q2: Day of a Week

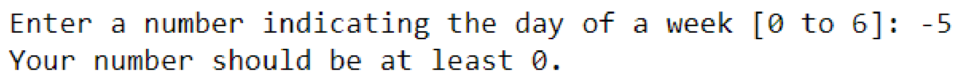
a) Assume the days of a week are numbered 0, 1, 2, 3, 4, 5 and 6 from Sunday to Saturday. Write a function called `get_day_of_week` which takes in a number between 0 and 6 (both inclusive) and returns a string that represents the corresponding day of a week. For example, if the number passed to the function is `4`, the function should **return** `'Thursday'`.

b) Prompt the user for a number between 0 and 6 (both inclusive). Assume that the user is always going to enter an integer. Call the function above and then display on the screen the corresponding day of a week.

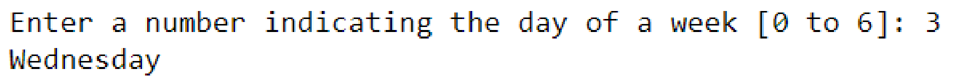
   a. If the user enters a number below 0, display the following error message: `'Your number should be at least 0.'`

   b. If the user enters a number above 6, display the following error message: `'Your number should be at most 6.'`

Two sample runs of the code can be found below:



```
Enter a number indicating the day of a week [0 to 6]: -5
Your number should be at least 0.
```



```
Enter a number indicating the day of a week [0 to 6]: 3
Wednesday
```

Q3: Tax Calculator

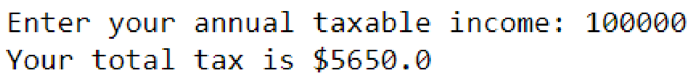
Recall that last week we had an exercise on tax calculation. This week we will use `if`/`else` to solve the problem, which is much easier than using the `max()` function to solve the problem.

Refer to the table below for the income tax rates:

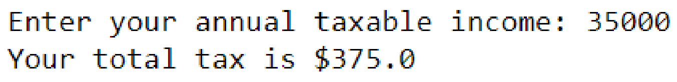
Chargeable Income	Income Tax Rate (%)	Gross Tax Payable ($)
First $20,000 Next $10,000	0 2	0 200
First $30,000 Next $10,000	- 3.50	200 350
First $40,000 Next $40,000	- 7	550 2,800
First $80,000 Next $40,000	- 11.5	3,350 4,600
First $120,000 Next $40,000	- 15	7,950 6,000
First $160,000 Next $40,000	- 18	13,950 7,200
First $200,000 Next $40,000	- 19	21,150 7,600
First $240,000 Next $40,000	- 19.5	28,750 7,800
First $280,000 Next $40,000	- 20	36,550 8,000
First $320,000 In excess of $320,000	- 22	44,550

a) Write a function called `calculate_income_tax()` that takes in the annual taxable income of a person and returns the total tax the person has to pay. The function should be able to handle any income value of 0.0 and above. Use conditional statements to help you with the implementation.

b) Prompt the user for his/her annual taxable income and display the tax he/she has to pay. Two sample runs are shown below.



```
Enter your annual taxable income: 100000
Your total tax is $5650.0
```



```
Enter your annual taxable income: 35000
Your total tax is $375.0
```

Q4: Mooncakes

A shop that sells mooncakes offers discount based on how many boxes of mooncakes a customer is going to buy.

- The shop sells the following two brands of mooncakes:
  - Tung Lok: $55.40 / box
  - Man Fu Yuan: $59.60 / box

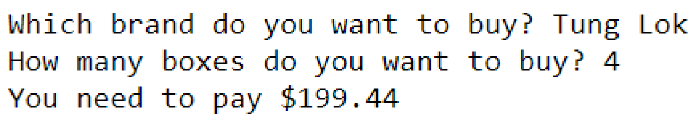
- If a customer buys 5 or more boxes of mooncakes of the same brand, a discount of 20% will be applied.
- If a customer buys between 2 and 4 boxes of mooncakes of the same brand, a discount of 10% will be applied.
- If a customer buys only 1 box of mooncakes, no discount is applied.

a) Write a function called `get_discount_rate`. This function takes in a parameter called `num_boxes` that indicates how many boxes of mooncakes of a particular brand a customer wants to buy. The function returns the discount rate. For example, if `num_boxes` is 6, the function returns 0.2. The function returns 0.0 if no discount is offered.

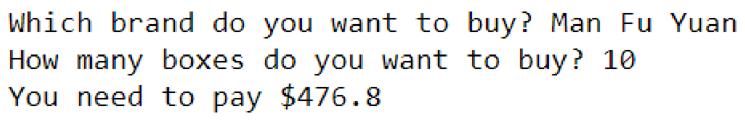
b) Write another function called `calculate_total_amount`. This function takes in two parameters: (1) `brand`, which is a string indicating the brand of mooncakes, and (2) `num_boxes`, which indicates the number of boxes a customer wants to buy. The function returns the total amount the customer has to pay for buying that number of boxes of mooncakes of that brand.

c) Prompt the customer for the brand of mooncakes he/she wants to buy and the number of boxes he/she wants to buy. Display the amount the customer has to pay. You can assume that the customer is always going to enter either "Tung Lok" or "Man Fu Yuan" as the brand and an integer as the number of boxes.

Two runs of the program are shown below:



```
Which brand do you want to buy? Tung Lok
How many boxes do you want to buy? 4
You need to pay $199.44
```



```
Which brand do you want to buy? Man Fu Yuan
How many boxes do you want to buy? 10
You need to pay $476.8
```

Q5: Honey

This exercise is to be done with Notepad++ and Anaconda Prompt.

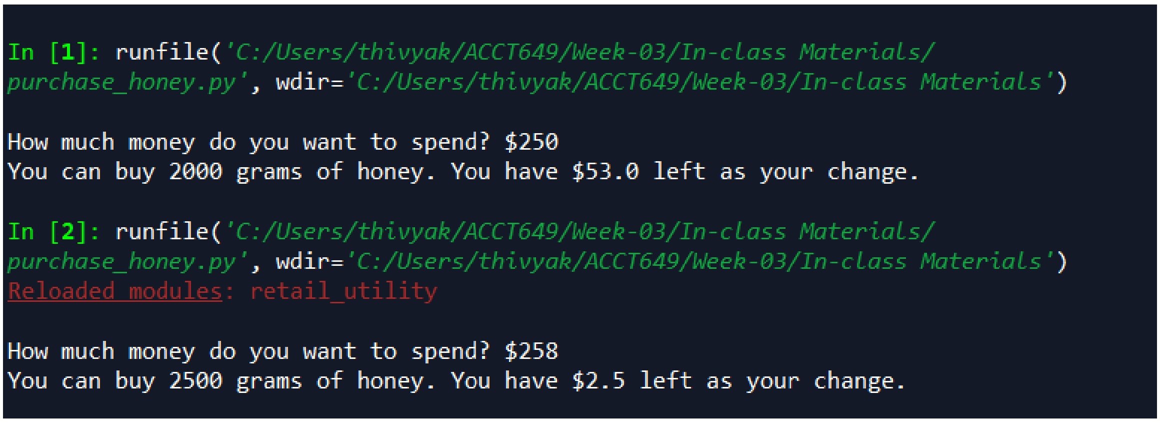
A shop sells honey in two sizes of jars. A 500g jar is priced at $58.50, while a 1kg jar of honey is priced at $98.50.

i. Inside a file called **`retail_utility.py`**, write a function called `calculate_max_quantity_and_change`. This function takes in two parameters:
   a. `unit_price`: The unit price of an item (e.g., $58.50).
   b. `amount`: The total amount of money a customer wants to spend to buy that item.

   This function returns a tuple that contains two values: (1) The maximum quantity of that item the customer can buy with the specified amount of money. (2) The change (remaining amount of money) the customer has after buying the maximum quantity of that item. For example, if `unit_price` is $58.50 and `amount` is $130.00, then the function should return 2 as the maximum quantity and $13.00 as the change, and hence it should return the tuple `(2, 13.0)`.

ii. Inside another file called **`purchase_honey.py`**, prompt the user for the amount of money he/she wants to spend to buy honey from this shop. Honey can be bought in 1kg jars (which cost $98.50) and 500g jars (which cost $58.50). Assume that the user wants to maximize the total amount of honey he/she could buy. Use the function you've implemented above to calculate the total amount of honey he/she could buy and the remaining amount of money he/she has. Use the function you have implemented in **`retail_utility.py`** to help you.

Two sample runs can be found below:



```
In [1]: runfile('C:/Users/thivyak/ACCT649/Week-03/In-class Materials/
purchase_honey.py', wdir='C:/Users/thivyak/ACCT649/Week-03/In-class Materials')

How much money do you want to spend? $250
You can buy 2000 grams of honey. You have $53.0 left as your change.

In [2]: runfile('C:/Users/thivyak/ACCT649/Week-03/In-class Materials/
purchase_honey.py', wdir='C:/Users/thivyak/ACCT649/Week-03/In-class Materials')
Reloaded modules: retail_utility

How much money do you want to spend? $258
You can buy 2500 grams of honey. You have $2.5 left as your change.
```

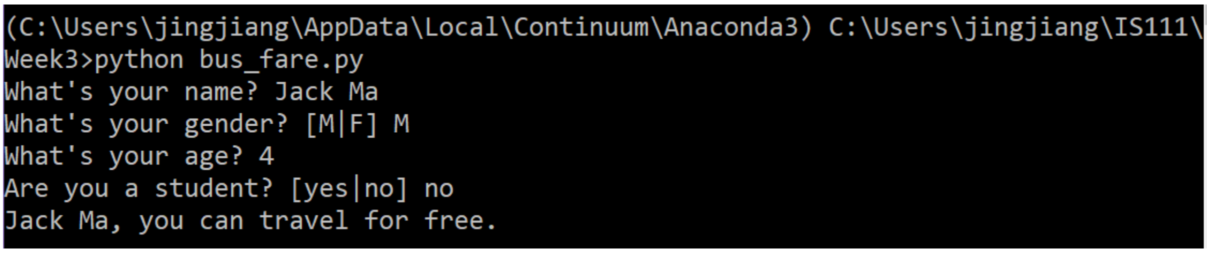
Q6: Bus Fares

We have provided you with a script called `bus_fare.py`. Inside the file we have implemented a function called `get_user_info`. Do not modify this function.

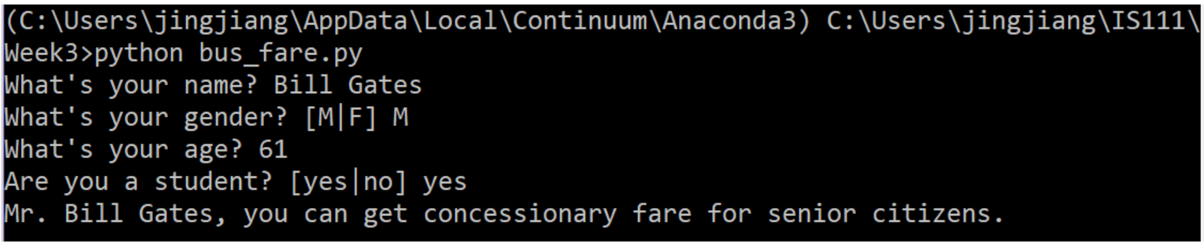
Write code in this file to do the following:

- Use the function `get_user_info` to obtain some information from the user.
- Based on the users' information, display the following messages accordingly:
  - If the user is 6 years old or below, display "`<name>, you can travel for free.`"
  - If the user is 60 years old or above, display "`<Mr.|Ms.> <name>, you can get concessionary fare for senior citizens.`"
  - If the user is above 6 and below 60, and the user is a student, display "`<Mr.|Ms.> <name>, you can get concessionary fare for students.`"
  - For all the other users, display "`<Mr.|Ms.> <name>, you need to pay full fare.`"

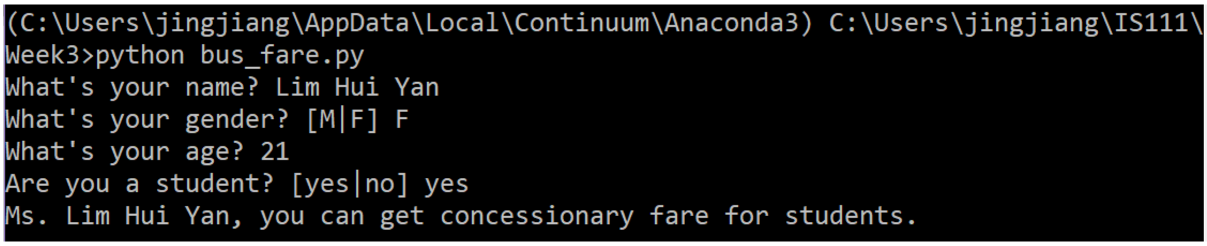
Three sample runs of the program can be found below:



```
(C:\Users\jingjiang\AppData\Local\Continuum\Anaconda3) C:\Users\jingjiang\IS111\
Week3>python bus_fare.py
What's your name? Jack Ma
What's your gender? [M|F] M
What's your age? 4
Are you a student? [yes|no] no
Jack Ma, you can travel for free.
```



```
(C:\Users\jingjiang\AppData\Local\Continuum\Anaconda3) C:\Users\jingjiang\IS111\
Week3>python bus_fare.py
What's your name? Bill Gates
What's your gender? [M|F] M
What's your age? 61
Are you a student? [yes|no] yes
Mr. Bill Gates, you can get concessionary fare for senior citizens.
```



```
(C:\Users\jingjiang\AppData\Local\Continuum\Anaconda3) C:\Users\jingjiang\IS111\
Week3>python bus_fare.py
What's your name? Lim Hui Yan
What's your gender? [M|F] F
What's your age? 21
Are you a student? [yes|no] yes
Ms. Lim Hui Yan, you can get concessionary fare for students.
```

Q7: Taxi Fares Revisited

[*challenging] We are going to improve the taxi fare calculator we implemented in class.

The real taxi fare structure is fairly complex. Refer to the following link for how taxi companies calculate taxi fares in Singapore:

https://www.lta.gov.sg/content/ltaweb/en/public-transport/taxis%20and%20private%20hire%20cars/fares-and-payment-methods.html

Because the real fare structure is so complex, we will use the following simplified fare structure for this question.

- The total taxi fare of a ride consists of two parts: (1) meter fare, and (2) surcharges.
- The meter fare of a ride consists of two parts: (1) flag-down fare, and (2) fare based on distance rate.
  - The flag-down fare convers the first **1km** of a ride. If the ride is shorter than 1km, the total meter fare is the flag-down fare.
  - If the ride is between **1km** and **9.8km**, the additional distance beyond 1km incurs a fare based on a distance rate per 400 meters. (Note that we use 9.8km instead of 10km here to simplify the calculation later.)
  - For any additional distance beyond **9.8km**, there is another distance rate per 350 meters.
- We consider two kinds of surcharges: (1) time-based surcharges, and (2) location surcharges.
  - We assume that if the ride is during a peak period, the time-based surcharge is 25% of the meter fare. If a ride is between midnight and 6am, the time-based surcharge is 50% of the meter fare. Other rides don't have any time-based surcharge.

For example, suppose we have the following rates for a taxi ride:

- Flag-down: $3.50
- Beyond 1km and up to 9.8km: 22 cents per 400 meters
- After 9.8km: 22 cents per 350 meters

Suppose a taxi ride has a distance of 11400 meters. Also suppose that the ride is between midnight and 6am, and there is a location surcharge of $3.00.

Then the total taxi fare is calculated as follows:

- Meter fare:
    - Flag-down for the first 1km: $3.50
    - Next 8800 meters: $0.22 * 22 = $4.84
    - Next 1600 meters: $0.22 * 5 = $1.10
    - Total meter fare: $9.44
- Time-based surcharge:
    - $9.44 x 50% = $4.72
- Location surcharge:
    - $3.00
- Total fare: ($9.44 + $4.72 + $3.00) = $17.16

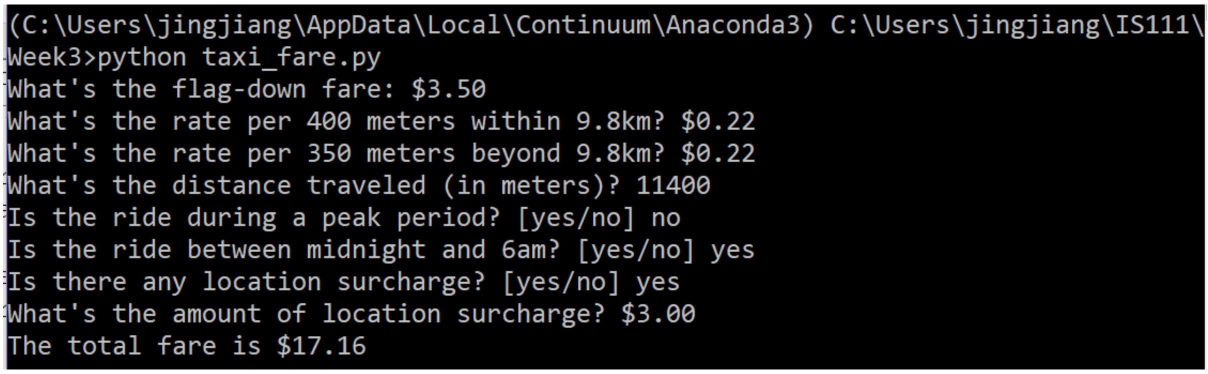
Create a python script called `taxi_fare.py` that helps a user calculate the taxi fare of a journey.

Two sample runs of the program can be found below.

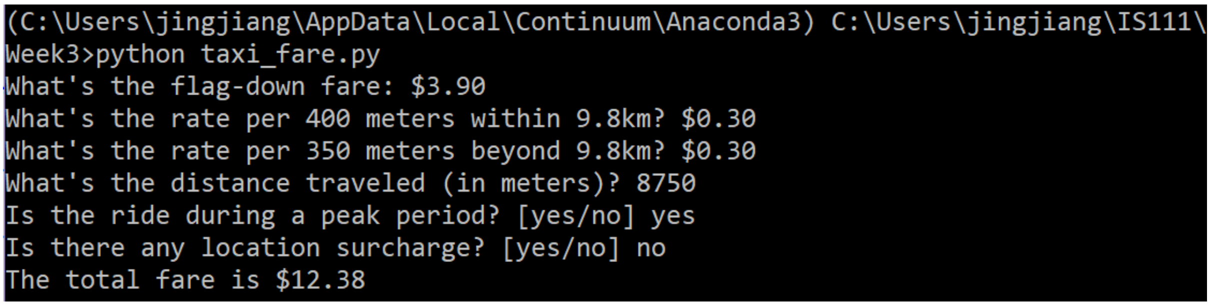
Note:

- For the distance travelled, you can assume that the user is always going to enter a whole number.
- If the ride is during a peak period, the program should not ask the user whether the ride is between midnight and 6am.

Design function(s) to help you in your implementation.



```
(C:\Users\jingjiang\AppData\Local\Continuum\Anaconda3) C:\Users\jingjiang\IS111\
Week3>python taxi_fare.py
What's the flag-down fare: $3.50
What's the rate per 400 meters within 9.8km? $0.22
What's the rate per 350 meters beyond 9.8km? $0.22
What's the distance traveled (in meters)? 11400
Is the ride during a peak period? [yes/no] no
Is the ride between midnight and 6am? [yes/no] yes
Is there any location surcharge? [yes/no] yes
What's the amount of location surcharge? $3.00
The total fare is $17.16
```



```
(C:\Users\jingjiang\AppData\Local\Continuum\Anaconda3) C:\Users\jingjiang\IS111\
Week3>python taxi_fare.py
What's the flag-down fare: $3.90
What's the rate per 400 meters within 9.8km? $0.30
What's the rate per 350 meters beyond 9.8km? $0.30
What's the distance traveled (in meters)? 8750
Is the ride during a peak period? [yes/no] yes
Is there any location surcharge? [yes/no] no
The total fare is $12.38
```