Github Link :
https://github.com/rmuppala/PythonClass/wiki/Lab2
https://github.com/rmuppala/PythonClass/tree/master/Lab2

Author
Name: Trinada Rajeswari Muppala, Class Id: 32

Objective:
Problem 1 – Identify the books can buy with in range

Features
Functions, main, dictionary, int

Configuration
Install Python , pyCharm

Input
Prompt user to enter the books , use for look exit when user entered 'N' to continue
{'Chemistry': '42', 'Math': '66', 'Java': '33', 'Web': '22', 'Physics': '36'}

Output

Books can buy in range 35   50   ['Chemistry', 'Physics']

Full output

```
Enter book name
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['C:\\Users\\mraje_000\\Documents\\pythonclass\\Python_Lesson4_SourceCode',
'C:/Users/mraje_000/Documents/pythonclass/Python_Lesson4_SourceCode'])
Java
Enter price : >? 33
Do you want to continue 'Y/N' >? Y
Enter book name  >? Web
Enter price : >? 22
Do you want to continue 'Y/N' >? Y
Enter book name  >? Physics
Enter price : >? 36
Do you want to continue 'Y/N' >? Y
Enter book name  >? Chemistry
Enter price : >? 42
Do you want to continue 'Y/N' >? Y
Enter book name  >? Math
Enter price : >? 66
Do you want to continue 'Y/N' >? N
What is the minimum price you can a buy a book >? 35
What is the maximum price you can a buy a book >? 50
Books Details  {'Chemistry': '42', 'Math': '66', 'Java': '33', 'Web': '22', 'Physics': '36'}
Books can buy in range  35   50   ['Chemistry', 'Physics']
PyDev console: starting.
```

Explain the code

1. Define function pass books details , min , max range and get books has price within the range of min and max value, return books list fall in that range

```
def bookscanbuy( books , minRange, maxRange):
 bookslist = []
 print("Books Details " , books)
 for bookname in books.keys() :
     bookvalue = books[bookname]
     if ( bookvalue >= minRange and bookvalue <=  maxRange):
       bookslist.append(bookname)
 return bookslist
```

2. Create main
3. Start with cont is 'Y'
4. Define dictionary
5. Use while loop and get all the book names and price, continue the loop until user entered 'N'

```
def main():
    cont='Y'
    books = dict()
    while cont == 'Y':   #Continue read book until user say 'N'
        book = input("Enter book name  " )
        price = input("Enter price : " )
        books[book] = price  #Create books dictionary
        cont = input("Do you want to continue 'Y/N' ")
```

5. Prompt user to provide min range and max range

```
min = input("What is the minimum price you can a buy a book ")
max = input("What is the maximum price you can a buy a book ")
```

6. Get books names got price with in min , max range by calling bookscanbuy function and print books fall in that range

```
blist = bookscanbuy(books, min,max)
print("Books can buy in range " ,  min, " " , max , " ", blist)
```

Explain about the deployment

Just create run/debug configuration

Limitation:

- Main program will not execute automatically need to `if __name__ == "__main__":`
- Books price, min range, max range are dynamic type if user entered non numeric value will get run time error

References –online tutorials, python class slides

Problem 2 – Modify contact number for one user

Features : functions, main, dictionary, len

Configuration: Install Python , pyCharm

Input
Users details :
[{'name': 'Mary', 'email': 'mary@gmail.com', 'number': '9875674322'}, {'name': 'Tania', 'email': 'tania@mail.umkc.edu', 'number': '8889765431'}, {'name': 'Robert', 'email': 'robert@hotmail.com', 'number': '7654321890'}, {'name': 'Bill', 'email': 'bill@hotmail.com', 'number': '9345678291'}]

Output

User provided user name Robert to change contact number and new contact number

Robert phone number changed from 7654321890 to 9845676544

[{'name': 'Mary', 'email': 'mary@gmail.com', 'number': '9875674322'}, {'name': 'Tania', 'email': 'tania@mail.umkc.edu', 'number': '8889765431'}, {'name': 'Robert', 'email': 'robert@hotmail.com', 'number': '9845676544'}, {'name': 'Bill', 'email': 'bill@hotmail.com', 'number': '9345678291'}]

Full output

```
Running C:/Users/mraje_000/Documents/pythonclass/Python_Lesson4_SourceCode/Lab2_2.py
Enter username
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['C:\\Users\\mraje_000\\Documents\\pythonclass\\Python_Lesson4_SourceCode',
'C:/Users/mraje_000/Documents/pythonclass/Python_Lesson4_SourceCode'])
Mary
Enter contat number : >? 9875674322
Enter email id>? mary@gmail.com
Do you want to continue 'Y/N' >? Y
Enter username  >? Tania
Enter contat number : >? 8889765431
Enter email id>? tania@mail.umkc.edu
Do you want to continue 'Y/N' >? Y
Enter username  >? Robert
Enter contat number : >? 7654321890
Enter email id>? robert@hotmail.com
Do you want to continue 'Y/N' >? Y
Enter username  >? Bill
Enter contat number : >? 9345678291
Enter email id>? bill@hotmail.com
Do you want to continue 'Y/N' >? N
Enter contact name to modify contact number>? Robert
Enter new contact number for Robert>? 9845676544
Users details :  [{'name': 'Mary', 'email': 'mary@gmail.com', 'number': '9875674322'}, {'name':
'Tania', 'email': 'tania@mail.umkc.edu', 'number': '8889765431'}, {'name': 'Robert', 'email':
'robert@hotmail.com', 'number': '7654321890'}, {'name': 'Bill', 'email': 'bill@hotmail.com', 'number':
'9345678291'}]
```

New user details after modifying phone number : [{'name': 'Mary', 'email': 'mary@gmail.com', 'number': '9875674322'}, {'name': 'Tania', 'email': 'tania@mail.umkc.edu', 'number': '8889765431'}, {'name': 'Robert', 'email': 'robert@hotmail.com', 'number': '9845676544'}, {'name': 'Bill', 'email': 'bill@hotmail.com', 'number': '9345678291'}]
PyDev console: starting.

Explain the code

1. Define function pass users list, name of the user contact to change and contact number

```
def modifyContact(contList, name, phno):
 for cont in contList:
     if(cont["name"] == name):
         cont["number"] = phno
```

2. Create main
3. Start with cont is 'Y'
4. Use while loop and get all user details name, number, email continue the loop until user entered 'N'. Each user details create as dictionary and add to users list

```
def main():
cont='Y'
contList = list()

while cont == 'Y':   #Read user details
    contDetl = dict() #each user details
    name = input("Enter username  " )
    phno = input("Enter contat number : " )
    if(len(phno) != 10): #exit if phone number is not 10 digits
        print("Phone number should be 10 digit")
        exit()
    email = input("Enter email id")
    contDetl["name"] = name
    contDetl["number"] = phno
    contDetl["email"] = email
    contList.append(contDetl) #create users list
    cont = input("Do you want to continue 'Y/N' ")
```

5. Get the name of the user and contact number to modify

```
contName = input("Enter contact name to modify contact number")
newph = input("Enter new contact number for " + contName)
```

6. Print users list, Update user contact by calling the function `modifyContact` and print the new contacts list

```
print("Users details : " , contList)
modifyContact(contList, contName, newph)
print("New user details after modifying phone number : " , contList)
```

Explain about the deployment

Just create run/debug configuration

Limitation:

- Main program will not execute automatically need to `if __name__ == "__main__":`
- Python function argument is call by Reference, needs to be careful. Original list is getting modified

References –online tutorials, python class slides

Problem 3 – Define Airline Booking Reservation System

Features : Class, inheritance

Configuration: Install Python , pyCharm

Input:

Define 3 person objects, one employee, one passenger, one flight, one food.
Passenger inheritance employee and flight, as element food

Output

Passenger Name : Bill, Age : 29, Contact : 9765478908
Traveling in flight : AI120 From :Denver To: Kansas Date : 02/15/2018 Dep : 12:30 AM Arr :4:30 PM
FoodOrdered : Lunch, Veg, price 0

Explain the code

1. Define Flight Class

```
class Flight(object): #class 1

  def __init__(self, no, frm, tol ,date , dep, arr):
      self.flightNo = no
      self.source = frm
      self.destination = tol
      self.flightDate = date
      self.departure = dep
      self.arrival = arr
```

2. Define Person class

```
class Person(object): #class 2

    def __init__(self,name, age, contact):
        self.name  = name
        self.age = age
        self.phno = contact
```

3. Define Employee class inheritance Person

```
class Employee(Person): # class 3 single inheritance

    def __init__(self,per, salary, title):
        Person.__init__(self,per.name, per.age, per.phno)
        self.salary = salary
        self.title = title
```

4. Define Food class

```
class Food(object): #class 4
    def __init__(self, meal, type, price):
        self.meal = meal
        self.type = type
        self.price = price
```

5. Define Passenger class , multiple inheritance from Person and Flight. Set Food details using foodOrdered function. Define passengerDetails() to Print passenger personal details, flight details and food ordered

```
class Passenger(Person, Flight): #class 5 multiple inheritance Person,
                                 Flight

        def __init__(self,person, flight): #constructor passing person
                                           flight object

            Person.__init__(self,person.name, person.age, person.phno)
            Flight.__init__(self, flight.flightNo, flight.source,
flight.destination ,flight.flightDate , flight.departure, flight.arrival)


        def foodOrdered(self, food): # setting food object
            self.food = food

        def passengerDetails(self):
            print("\nPassenger Name : " + self.name + ", Age : " + self.age +
", Contact : " + self.phno )
            print("Traveling in flight : " + self.flightNo + " From :" +
self.source + " To: " + self.destination +  " Date : " + self.flightDate +
" Dep : " + self.departure + " Arr : " + self.arrival)
            print("FoodOrdered : ", self.food.meal + ", " + self.food.type + ",
price " + str(self.food.price))
```

6. In Main define 3 person objects , 2 food objects, 2 employees, one passenger

```
def main():
    person1 = Person("Tina", "32", "5673891730") #instance of Person class
    person2 = Person("Robert" , "45" , "9234567890")
    person3 = Person("Bill", "29", "9765478908")

    flight1 = Flight("AI120", "Denver", "Kansas" ,"02/15/2018" , "12:30 AM",
"4:30 PM") #instance of Flight class

    food1 = Food("Breakfast", "snack", 3) #instance of Food class
    food2 =  Food("Lunch", "Veg", 0)

    employee1 = Employee(person1,50000,"Crew") #instance of Employee class
    employee2 = Employee(person2,75000,"Pilot")
```

7. Define passenger constructor pass person3 and flight 1

```
passenger1 = Passenger(person3,flight1) #instance of Passenger class

passenger1.foodOrdered(food2)

passenger1.passengerDetails()


if __name__ == "__main__":
    main()
```

Explain about the deployment

Just create run/debug configuration

Limitations:

In Python can't define type of the fields. Print statement for food price errored at run time as int can't be print. Python don't give compilation errors. This will be difficult when processing large data

Don't get compilation error if referring wrong attribute name for the superclass. In passenger I have tried person.contact but the actual field is phno. Got error at runtime

References –online tutorials, python class slides

Problem 4 – Create 15 random numbers between 0 to 20 and print most frequest number in the list

Features : numpy

Configuration: Install Python , pyCharm

Input

Random numbers generated using numpy
[7 8 5 8 17 16 19 0 15 8 16 6 7 18 12]

Output

Frequest number in the list:
8

Explain the code

1. Import numpy, and generate random integer numbers

```
import numpy as npy
rnums = npy.random.randint(0, 20, 15)
print("Numbers List:")
print(rnums)
```

2. Call numpy bincount method Counts number of occurrences of each value in array, call argmax which provides the maximum value which number occurs the most

```
print("Frequest number in the list:")
print(npy.bincount(rnums).argmax())
```

Explain about the deployment

Just create run/debug configuration

Limitation:

Just learning numpy don't know much limitation.

References –online tutorials, python class slides