University of Windsor

Lab 2

Group - 1	
Chintankumar Patel	110093036

Lab2-Writing a Complete Script in Python

1. Write a Python program that prompts the user to enter a date in the format "MM/DD/YYYY", and then converts it to a datetime object

```
from datetime import datetime
# user input in specif fromat
while True:
    try:
        date = input("Please enter the date in this format MM/DD/YYYY")
        print(date)
        # used strptitme method to convert string into datetime obj
        datetimeobj = datetime.strptime(date, "%m/%d/%Y")
        break
    except ValueError:
        print("Please add in this format MM/DD/YYYY")
        continue
```

2. Write a Python function that takes a datetime object and formats it as a string in the format "YYYY-MM-DDTHH:MM:SSZ"

```
#YYYY-MM- DDTHH:MM:SSZ
# convert datetime obj into string using strftime method
def objtostr(datetimeobj):
    fromatedString = datetimeobj.strftime("%Y-%m-%dT%H:%M:%SZ")
    return fromatedString

#current time
currentTime = datetime.now()
#fun calling
formatedString = objtostr(currentTime)
print("Date and Time in String Format",formatedString)
```

3. Write a Python function that takes a datetime object and returns the date and time as separate strings.

```
# spliting date and time from object
def separateDateTime(datetimeobj):
    date = datetimeobj.date()
    time = datetimeobj.time()
    return date, time

# using current time
date, time = separateDateTime(datetime.now())
print("Date", date)
print("Time", time)
```

4. Write a Python program that prompts the user to enter two dates in the format "MM/DD/YYYY", and calculates the number of days between them using the datetime library.

```
from datetime import datetime
def calculateDiff(date1, date2):
    date = date1-date2
    return date
while True:
    try:
        # userInputs
        date1 = input("Please enter the date in this format MM/DD/YYYY")
        date2 = input("Please enter the date in this format MM/DD/YYYY")
        datetimeobj1 = datetime.strptime(
            date1, "%m/%d/%Y")  # convert in str to obj
        datetimeobj2 = datetime.strptime(date2, "%m/%d/%Y")
        # calling fun for difference
        difference = calculateDiff(datetimeobj1, datetimeobj2)
        calcDays = difference.days  # only giving difference in days
        print("differenece between two dates is ", calcDays, " days")
        break
    except ValueError:  # handling exceptiom
        print("Please add in this format MM/DD/YYYY")
        continue
```

5. **Write a Python function that takes a datetime object and adds a specified number of days to it.**

```
def addDate(date, days):
    newDate = date + timedelta(days=days) # Adding extra day into current date
    return newDate

currentDate = datetime.now()
newDate = addDate(currentDate, 10)
print("Updated Date", newDate.date())
```

Part 2: Write a complete Python script, with comments, to do the following

6. Open a text file called "*catalog.txt*", attached with this lab, for reading. The file contains the items available in a fitness studio, the items categories/classes, and their quantities.

```
# Absolute path to file in reading mode
try:
    f = open('/Users/chintanpatel/Desktop/SEM3/Distributed/LAB/cataluog.txt', 'r')
    print(f.read())
except FileNotFoundError:
    print("File not found")
```

7. Define a list of strings called ***fit_items***. The list should contain at least 10 strings and each string represent a specific fitness item, e.g., treadmill, lifting bars, weights, etc.

```
fit_items = [
       "treadmill",
       "elliptical trainer",
       "exercise bike",
       "rowing machine",
       "weights",
       "dumbbells",
       "barbell",
       "smith machine",
       "leg press machine",
       "cable crossover machine"
       ]
```

8. Loop over each element in ***fit_items*** and check if that element matches any of the products in the file

```
# Absolute path to file in reading mode
file = open('/Users/chintanpatel/Desktop/SEM3/Distributed/LAB/catalog.txt', 'r')
newList = []
#iterating the file
for line in file:
    if line.strip() in fit_items: #check if it is available in above list
        newList.append(str(line.strip())) # if yes, add into newList
print(len(newList))
```

9. If there is a match, save the category and the quantity corresponding to that item in some variables
10. Create a dict ***d1*** with entries ***item:category*** where ***item*** **(key)** is the item (string) found in *catalog.txt* and ***category*** (**value**) is the category of that item. Add the item and its category to ***d1*** as {***item:category***}. Create another dict ***d2*** with entries ***item:quantity*** and add the item found and its quantity to ***d2***.

```
file = open('/Users/chintanpatel/Desktop/SEM3/Distributed/LAB/catalog.txt','r')
#define dict
d1={}
d2={}

#read all line and convert it into list l
l = file.readlines()

#iterate till the lenfth of l
for i in range(0,len(l)):
    if l[i].strip() in fit_items: #if it is available in above list
        item = l[i].strip() #get item name
        category = l[i+1].strip()  #get category of item
        quantity = l[i+2].strip()  #get quantity of item
```

```
        d1.update({item:category}) #udpate dict
        d2.update({item:quantity}) #update dict
```

11. Next the program should ask the user to enter a string ***s, representing a fitness item,*** as an input and the *category* of ***s*** from ***d1*** and the *quantity* from ***d2*** .
12. If the item's name entered by the user does not correspond to a valid key, the program should catch an exception. When the exception occurs, display an appropriate error message then prompt the user to input another item's name.

```
#infinte loop
while True:
try:
       userInput = input("Enter the fitness item ") #taking user input
       print("Category: ", d1[userInput]) #display cat and quantity
       print("Quantity: ", d2[userInput])
except KeyError: # if userinput not available in dict
       print("We do not any any item")
userInput2 = input("Do you want to continue (Yes/No) ") #Userinpput
if (userInput2.lower() == "no"): #if no then break the loop and exit program
       break
elif (userInput2.lower() == "yes"): #if yes then continue with userinput
       continue
else:
       print("Please select only (Yes/No) ") #if wront input then asking again
```

Part 3: Answer the following questions:

1. Assume that you have a list named L, e.g., L = [19, 52, 87, 2, 8, 11, 18, 22], write a Python script to count the number of odd numbers in L?

```
L = [19, 52, 87, 2, 8, 11, 18, 22]
oddList = [] #to add ODD value
for l1 in L: #itrate loop
if (l1 % 2 != 0): #if modulo is not eqaul to zero
       oddList.append(l1) # adding to list
else:
       continue  #if equal to zero continue to iterate
print("The total odd numbers in L is ", len(oddList))
```

2. Assume that you have a list named L, e.g., L = [38, 5, 7, 2, 8,112,18,400], write a Python script to calculate the average of all even numbers in L?

```
L = [38, 5, 7, 2, 8,112,18,400]
evenList = []  # to add EVEN value
number=0

for l1 in L:
```

```
    if(l1%2==0): #if modulo is zero then add into list
        evenList.append(l1)
    else:
        continue #if not, continue iterete

for evenNum in evenList: #iterete even list
    number += evenNum #sum the all value

#divide with the total even number
print("Average value of all even number in List",number/len(evenList))
```

3. Assume that you have a list named L, e.g., L = [38, 5, 7, 2, 8,112,18,400], write a Python script to find the largest and the smallest number in L?

```
import sys
L = [38, 5,7, 2, 8,112,18,400]

#assign max and min value
minValue=sys.maxsize
maxValue= -sys.maxsize

for l1 in L:
    if(l1<minValue): #if each value is less then minvalue or not
        minValue=l1  #if true, change minvalue
    else:
        continue  #if not, continue iterete
for l2 in L:
    if(l2>maxValue):  #if each value is less then maxvalue or not
        maxValue=l2 #if true, change maxvalue
    else:
        continue #if not, continue iterete

print("Smallest Number in List is",minValue)
print("Largest Number in List is",maxValue)
```