ASSIGNMENT 2.2

December 4, 2020

NAME: SHUBHAM TAKANKHAR

CLASS SY MCA

ROLL NO 54

0.1 1.WAP that has class person storing Name and DOB of the person. The program should subtract DOB from todays date and find out whether a person is eligible to vote or not?

[15]:
```
import datetime
now = datetime.datetime.now()
class person:
    def __init__(self,name,dob):
        self.name=name
        self.dob=dob
    def eligibility(self):
        if (now.year - self.dob > 18 ):
            print("{} is eligible for voting".format(self.name))
        else:
            print("{} is not eligible for voting{}".format(self.name,self.dob))

shubh=person("shubham",1997)
shubh.eligibility()
```

```
shubham is eligible for voting
```

0.2 2.WAP that has a class store which keeps a record of code and price of each product. Display a menu of all the products to the user and prompt him to enter the quantity of each item required. Generate a bill and display the total amount.

[34]:
```
class store:
    def __init__(self):
        self.products={"pen":10,"pencil":"5","eraser":2,"sharpner":4,"scale":
↪10,"compass":200,"notebook":50}
    def display(self):
        print ("{:<10} {:<10} ".format('Product', 'Price'))
        print("---------- ----------")
        num = 1
        prods=['etc']
        for key, value in self.products.items():
            print ("{} {:<10} {:<10}".format(num,key, value))
            prods.append(key)
            num+=1
            usr='S'
            total=0
        print("\n")
        while(usr not in ['e','E']):
            prodNum=int(input("\n Enter Product No to add to cart:"))
            quantity=int(input("Enter the quantity of product (MAX 10):"))

            total += self.products[prods[prodNum]] * quantity

            usr=input("Press any key to continue or To Generate Bill Press E ")
        print("Your Total Bill is:"+str(total))

stationary = store()
stationary.display()
```

```
Product    Price
---------- ----------
1 pen        10
2 pencil     5
3 eraser     2
4 sharpner   4
5 scale      10
6 compass    200
7 notebook   50


Your Total Bill is:402
```

0.3 3.WAP to deposit or withdraw money in a bank account.

[6]:
```
class Bank:
    def __init__(self):
        self.balance = 0
        print ("The account is created")

    def deposit(self):
        amount = int(input("Enter amount to be deposited:"))
        self.balance += amount
        print("Successfully Deposited ! \n Your Updated Balance is:{}".
 ↪format(self.balance))

```

```
    def withdraw(self):
        amount = int(input(("Enter the amount to withdraw:")))
        if (amount > self.balance) :
            print("Insufficient Balance !")
        else:
            self.balance -= amount
            print("Successfully Withdrawn ! \n Updated Balance is:{}".
 ↪format(self.balance))

    def enquiry(self):
        print ("Balance in the account is "+ str(self.balance))


account = Bank()
account.deposit()
account.withdraw()
account.enquiry()
```

```
The account is created
Successfully Deposited !
 Your Updated Balance is:1000
Successfully Withdrawn !
 Updated Balance is:800
Balance in the account is 800
```

0.4 4.WAP that has a class point with attributes as x and y coordinates.Make two objects of this class and find the midpoint of both the points.

[8]:
```
class point:
    def __init__(self,x,y):
        self.x=x
        self.y=y

def midpoint(p1,p2):
    x= (p1.x+p2.x)/2
    y= (p1.y+p2.y)/2
    print("Midpoints are X:{} Y:{}".format(x,y))

p1=point(10,10)
p2=point(20,20)

midpoint(p1,p2)
```

```
Midpoints are X:15.0 Y:15.0
```

0.5 5.WAP that uses a class attribute to define some default titles for faculty in a college. Display neme along with title and department of the college.

```
[13]: class attribute:
          def __init__(self,name,title="Professor",dept="IT"):
              self.name=name
              self.title=title
              self.dept=dept

      name = input("Enter your name")

      shubh = attribute(name)

      print("NAME:{}\nTITLE:{}\nDEPARTMENT:{}".format(shubh.name,shubh.title,shubh.
       ↪dept))
```

```
NAME:shubham
TITLE:Professor
DEPARTMENT:IT
```

0.6 6.Write a class that has list of integers as data members. Read(),display, find_largest(), find_smallest() and find_mean() as its member functions.

```
[19]: class integers:
          def __init__(self,data):
              self.data=data
          def read(self):
              num = int(input("Enter Number:"))
              self.data.append(num)
          def display(self):
              print("List of elements:")
              for i in self.data:
                  print(i,end="|")
          def find_largest(self):
              print("\n"+str(max(self.data))+" is the largest element")
          def find_smallest(self):
              print(str(min(self.data))+" is the smalles element")
          def find_mean(self):
              print("Mean:"+str(round(sum(self.data)/len(self.data))))

      nums=[2,5,6,1,10]

      numbers = integers(nums)

      numbers.read() #5

      numbers.display()
numbers.find_largest()

numbers.find_smallest()

numbers.find_mean()
```

```
List of elements:
2|5|6|1|10|5|
10 is the largest element
1 is the smalles element
Mean:5
```

[ ]:

```

```