PRACTICAL FILE

PYTHON PROGRAMMING

Python Programming Lab

Submitted By

Name: Khushi Wadhawan

Roll No.: R2142210423

SAP ID: 500093673

B. Tech CSE 2nd Sem
School of Computer Science

Submitted to

Mr. Amit Singh

School of Computer Science, UPES

School of Computer Science

University of Petroleum and Energy Studies,

Dehradun – 248007: Uttarakhand

EXPERIMENT # 1:

Q1. Given an integer n perform the following conditional actions:

- *If n is odd, print Weird*
- *If n is even and in the inclusive range of 2 to 5 , print Not Weird*
- *If n is even and in the inclusive range of 6 to 20, print Weird*
- *If n is even and greater than 20, print Not Weird*

CODE:

```
n=int(input("enter a no."))
if (n%2==1):
    print("Weird")
else:
    if (n>=2 and n<=5):
        print("Not weird")
    elif (n>=6 and n<=20):
        print("Weird")
    elif (n>20):
        print("Not Weird")
```

Q2. WAP to read an integer 'n' from STDIN. For all non-negative integers i<n, print i^2 on a separate line.

CODE:

```
n=int(input("enter a no."))
for i in range(0,n,1):
    print(i*i)
```

Q3. WAP to read an integer from STDIN. Without using any string methods, print the following on a single line: 123…n

CODE:

```
n=int(input("enter a no."))
for i in range(1,n+1,1):
    print(i,end='')
```

EXPERIMENT # 2:

Q1. WAP to input a list of scores for N students in a list data type. Find the score of the runner-up and print the output. Sample Input N = 5 Scores= 2 3 6 6 5 Sample output 5 Note: Given list is [2, 3, 6, 6, 5]. The maximum score is 6, second maximum is 5. Hence, we print 5 as the runner-up score.

CODE:

Using max() and remove():

```
n=int(input("enter no. of students"))
l=[]
print("enter records")
for i in range(0,n,1):
    m=int(input())
    l.append(m)
l.remove(max(l))
print(max(l),"is the runner up")
```

Using comparision:

```
n=int(input("enter no. of students"))
l=[]
print("enter records")
for i in range(0,n,1):
    m=int(input())
    l.append(m)
smx,mx=l[0],l[0]
for j in range(0,n,1):
    if (l[j]>mx):
        mx=l[j]
for k in range(0,n,1):
    if (l[k]<mx):
        if (l[k]>smx):
            smx=l[k]

print(smx,"is the runner up")
```

Q 2. WAP for the following:

a. Check if a list contains an element

CODE:

```
n=int(input("enter no. of elements"))
print("enter elements")
l=[]
for i in range(0,n,1):
    k=int(input())
    l.append(k)
m=int(input("enter no. you want to search"))
for j in range(0,n,1):
            if(m==l[j]):
                print("element found at index",j)
```

b. How to iterate over 2+ lists at the same time

CODE:

```
name=['Drew','Lia','Rohan']
age=[30,24,35]
occupation=['salesman','artist','businessman']
z=zip(name,age,occupation)

for name,age,occupation in z:
    print("%s of age %d is a %s" % (name,age,occupation))
```

c. Show that the list mutable using program

CODE:

```
n=[1,2,3,4,5]
for i in range(0,5,1):
    n[i]=n[i]+1
print(n)
```

d. Distinguish using program in python:

i. extend vs append

CODE:

```
n=[1,2,3,4,5]
m=[6,7]
n.append(m)
print(n)
n.extend(m)
print(n)
```

ii. remove vs pop

CODE:

```
n=[1,2,3,4,5]
l=n.remove(4)
m=n.pop(3)
print(l)
print(m)
print(n)
```

e. remove duplicate elements from a list

CODE:

```
n=[1,2,3,4,5,3]
m=[]
for i in n:
    if i not in m:
        m.append(i)
print(m)
```

f. How to manipulate every element in a list with list comprehension

CODE:

```
n=[1,2,3,4,5,3]
m=[i+1 for i in n]
print(m)
```

g. Count the occurrence of a specific object in a list

CODE:

```
n=[1,2,3,4,5,3]
```

```
c=0
m=int(input("enter element occurence to be checked"))
for i in range(0,6,1):
    if (n[i]==m):
        c=c+1
print(" it occurs",c,"times")
```

Q 3. Python program to interchange first and last elements in a list.

CODE:

```
n=[1,2,3,4,5]
temp=n[0]
n[0]=n[4]
n[4]=temp
print(n)
```

Q 4. Write a program using for loop to find length of a string without len function and validate though the len function.

CODE:

```
n=[1,2,3,4,5]
c=0
for i in n:
    c=c+1
print(c)
print(len(n))
```

Q 5. Reverse a list in Python with in-build function and without in-build function

CODE:

Without function:

```
n=[1,2,3,4,5]
l=[]
for i in range(4,-1,-1):
    l.append(n[i])
print(n)
```

```
print(l)
```

with function:

```
n=[1,2,3,4,5]
n.reverse()
print(n)
```

Q 6. Extend nested list by adding the sublist

CODE:

```
n=[1,2,3,4,5]
l=[6,7]
for i in range(0,2,1):
    n.append(l[i])
print(n)
```

Q 7. Remove all occurrences of a specific item from a list

CODE:

```
n=[1,2,3,4,5,2,8,2]
for i in n:
    if i==2:
        n.remove(i)
print(n)
```

Q 8. Search for a specific element in the list and replace it with new element

CODE:

```
n=int(input("enter no. of elements"))
print("enter elements")
l=[]
for i in range(0,n,1):
    k=int(input())
    l.append(k)
m=int(input("enter no. you want to replace"))
```

```
for j in range(0,n,1):
            if(m==l[j]):
                l[j]=25
print(l)
```

Q 9. Write a Python program to multiply all the items in a list.

CODE:

```
n=[1,2,3,4,5]
m=1
for i in n:
    m=m*i
print(m)
```

Q 10. Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.

CODE:

```
n=['as','mango','roar','neon','app']
c=0
for i in n:
    if len(i)>2:
        if i[0]==i[len(i)-1]:
            c=c+1
print(c)
```

EXPERIMENT # 3:

Q 1. WAP to read the record of n students in a dictionary containing key/value pairs of the name. Print the average of the marks obtained by the particular student correct to 2 decimal places. For a program demonstration, consider at least records of five students.

CODE:

```
dict={}
n=int(input("enter no. of students"))
for i in range(n):
    name=input("enter student name")
    marks=int(input("enter student marks"))
    dict[name]=marks

s=0
for key,value in dict.items():
    s=s+value

print("the average is {:.2f}".format(s/n))
```

Q 2. Rupal has a huge collection of country stamps. She decided to count the total number of distinct country stamps in her collection. She asked for your help. You pick the stamps one by one from a stack of country stamps. a. Find the total number of distinct country stamps using a suitable data type. b. Count the number of stamps of the different countries. For a programming demonstration, consider at least 5 countries and multiple stamps of these five countries.

CODE:

```
s = []
n = int(input("Enter the total number of stamps: "))
c=0

for i in range(n):
    stamp = str(input("Enter the stamp value: "))
    s.append(stamp)

print("Total number of stamps are: ",n)

for i in s:
    for j in s:
        if i==j:
            c=c+1
```

```
    print("there are",c,i,"stamps")
    c=0
```

EXPERIMENT # 4:

Q1. WAP to enter a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left.

CODE:

```
n=input("enter a string")
m=input("enter a substring")
c=0
l=len(m)
for i in range(len(n)):
    if n[i:i+l]==m:
        c=c+1
print(c)
```

Q2. WAP to input the first name, middle and last name of a person. Your task is to print the initials of the first and middle name separated by a dot (.)

CODE:

```
n=input("enter first,middle and last name")
s=n[0].upper()
m=''
c=0
for i in range(len(n)):
    if n[i]==' ':
        s= s+'.'+n[i+1].upper()
        break
for i in range(len(n)):
    if n[i]==' ':
        c=c+1
        if c==2:
            for j in range(i+1,len(n)):
                m=m+n[j]
print(s+" "+m.capitalize())
```

Q3.Given a string containing both upper and lower case alphabets. Write a Python program to count the number of occurrences of each alphabet (case insensitive) and display the same.

CODE:

```
n=input("enter string")
m=n.upper()
l=list(m)
for i in range(len(l)):
    c=l.count(l[i])
    if l.index(l[i])== i:
                print(l[i]," -",c)
```

EXPERIMENT # 5 & 6:

Q1. Using functions, re-write and execute program to:

1. *Add natural numbers upto n, where n is taken as input from the user.*
2. *Print Fibonacci series till nth term and take input from the user.*

CODE:

Sum of natural numbers:

```
def sum_numbers(num):
    if(sum == 0):
        return(num)
    else:
        return(num*(num+1)/2)

n = int(input("enter a number : "))
sum = sum_numbers(n)
print("sum of natural numbers is : ", int(sum))
```

Fibonacci Series:

```
def fibonacci(n):
    if(n<=1):
        return (n)
    else:
        return(fibonacci(n-1) + fibonacci(n-2))

element  = int(input("Enter the range: "))
print("Fibonacci series is : ")
for i in range(element):
    print(fibonacci(i))
```

Q2 . At an airport, a traveler is allowed entry into the flight only if he clears the following checks:

1. Baggage Check
2. Immigration Check
3. Security Check

CODE:

```
def check_baggage(weight):
```

```
    if(weight >=0 & weight<=40):
        return (True)
    else:
        return (False)
def check_immigration(year):
    if(year>=2030 & year<=2050):
        return (True)
    else:
        return (False)
def check_security(noc):
    if(noc == "VALID" or noc == "valid"):
        return (True)
    else:
        return (False)
def Traveller():
    tid = int(input("Enter the traveller ID: "))
    n = input("Enter the name of traveller : ")
    weight = int(input("Enter the weight of baggage : "))
    year = int(input("Enter the expiry year : "))
    noc = input("Enter the status of NOC : ")
    if(check_baggage(weight) == True & check_immigration(year) ==
True & check_security(noc) == True):
        print("Traveller ID : ", tid)
        print("Traveller name : ", n)
        print("Allow the traveller to fly....")
    else:
        print("Traveller ID : ", tid)
        print("Traveller Name : ", n)
        print("Detain traveler for re-checking....")
ele = int(input("Enter the number of travellers : "))
for i in range(ele):
    Traveller()
```

Q3. Write a Python program to find the maximum and minimum values in a given list of tuples using lambda function.

CODE:

```
def Tuples(ls):
    Tmax = max(ls, key = lambda item:item[1])[1]
    Tmin = min(ls, key = lambda item:item[1])[1]
    return (Tmax, Tmin)

ls = [('V', 62), ('VI', 68), ('VII', 72), ('VIII', 70), ('IX', 74),
('X', 65)]
print("Original list of tuples: ")
print(ls)
print("Maximum and Minimum values from list of Tuples: ")
print(Tuples(ls))
```

EXPERIMENT # 7 &8:

Q1. Write a Python program to:
1. read a file.
2. add backslash (\) before every double quote in the file contents.
3. write it to another file in the same folder.
4. print the contents of both the files.

CODE:

```
str = ""
s1 = ""
with open("File1", "r") as nf:
    str = nf.readline()
    for i in range (0, len(str)):
        if str[i] == '"':
            s1 = s1+'\\'+str[i]
        else:
            s1 = s1+str[i]
print(str)
nf.close()
with open ("File2", "w") as nf:
    nf.write(s1)
    print(nf.readline)
nf.close
```

Q2. Consider a file 'rhyme.txt' in D Drive with following text:
Write a Python program to count the words in the file using a dictionary (use space as a delimiter). Find unique words and the count of their occurrences (ignoring case). Write the output in another file "words.txt" at the same location.

CODE:

```
f = open("File3", "r")
str = f.read()
f.close()
word = str.split()
print("Total number of words : ", len(word))
count = {}
for i in word:
    i = i.lower()
    if i in count:
        count[i] = count[i]+1
```

```
    else:
        count[i] = 1
for key in list(count.keys()):
    ele = print(key, ":", count[key])
```

Q3. Assume a file city.txt with details of 5 cities in given format (cityname population(in lakhs) area(in sq KM) .
Open file city.txt and read to:
a. Display details of all cities
b. Display city names with population more than 10Lakhs
c. Display sum of areas of all cities

CODE:

```
details = open("City", "r")
n = details.read()
print("Details of the cities are : ")
print(n)
details.close()
print("\n")

population = open("City", "r")
n = population.readlines()
population.close()
print("Cities with population more than 10 Lakh : ")
for i in n:
    i = i.split()
    if(int(i[1])>10):
        print(i[0])
print("\n")

sum = open("City", "r")
area = sum.readlines()
sum.close()
result = 0
for j in area:
    j = j.split()
    result = result + int(i[2])
print("Sum of are of all the cities : ", result)
```

EXPERIMENT: OBJECTS AND CLASSES:

CODE 1:

```
class Person(object):
    def __init__(self, name):
        self.name = name
    def getName(self):
        return self.name
    def isEmployee(self):
        return False
class Employee(Person):
    def isEmployee(self):
        return True
emp = Person("PERSON IS GOOD")
print(emp.getName(), emp.isEmployee())

emp = Employee("PERSON IS BAD")
print(emp.getName(), emp.isEmployee())
```

CODE 2:

```
class Person( object ):
        def __init__(self, name, idnumber):
                self.name = name
                self.idnumber = idnumber
        def display(self):
                print(self.name)
                print(self.idnumber)
class Employee( Person ):
        def __init__(self, name, idnumber, salary, post):
                self.salary = salary
                self.post = post
                Person.__init__(self, name, idnumber)
a = Employee('HARSH', 56980, 200000, "Intern")

# calling a function of the class Person using its instance
a.display()
```

CODE3:

```
class Parent():
       def first(self):
           print('first function')

class Child(Parent):
       def second(self):
          print('second function')

ob = Child()
ob.first()
ob.second()
```

CODE 4:

```
class Robot:
    def __init__(self, name):
        self.name = name
    def say_hi(self):
        print("Hi, I am " + self.name)
class PhysicianRobot(Robot):
    def say_hi(self):
        print("Everything will be okay! ")
        print(self.name + " takes care of you!")
y = PhysicianRobot("Arsh")
y.say_hi()
```

CODE 5:

```
class Parent:
      def func1(self):
          print("this is function 1")
class Child(Parent):
      def func2(self):
          print("this is function 2")
class Child2(Child):
      def func3(self):
          print("this is function 3")
ob = Child2()
ob.func1()
ob.func2()
ob.func3()
```