MCA 2nd Sem Python Lab Programs

1. Program to perform Arithmetic and Relational operations on given two Integer Numbers.

   Aim :-To demonstrate Arithmetic and Relational operators in Python using two integers.

   Algorithm :-

   Step1 : Start

   Step 2 : Enter and Read the two numbers in to ‘A’ and ‘B’

   Step 3 : Perform all arithmetic operations on ‘A’ and ‘B’

   Step 4 : Perform all relational operations on ‘A’ and ‘B’

   Step 5 : Stop

   Program:

```
#Arithmetic operations For integers

a=int(input("Enter the A value: "))
b=int(input("Enter the B value: "))

print("Addition of a and b",a+b)
print("Substraction of a and b",a-b)
print("Multiplication of a and b",a*b)
print("Power of a and b",a**b)
print("Remainder of a and b",a%b)
print("Quotient of a and b",a//b)
print("Division of a and b",a/b)

#Relational operations For integers
print(“The given A > B is”,a>b)
print(“The given A < B is”,a<b)
print(“The given A == B is”,a==b)
print(' The Gien A != B is',a!=b)
print('The Gien A >= B is',a>=b)
print('The Given A <= B is',a<=b)
```

Output1:

Enter the A value: 10

Enter the B value: 15

Addition of a and b 25

Subtraction of a and b -5

Multiplication of a and b 150

Power of a and b 1000000000000000

Remainder of a and b 10

Quotient of a and b 0

Division of a and b 0.6666666666666666

The given A > B is False

The given A < B is True

The given A == B is False

The Given A! = B is True

The Given A >= B is False

The Given A <= B is True

Output2:

Enter the A value: 15

Enter the B value: 12

Addition of a and b 27

Subtraction of a and b 3

Multiplication of a and b 180

Power of a and b 129746337890625

Remainder of a and b 3

Quotient of a and b 1

Division of a and b 1.25

The given A > B is True

The given A < B is False

The given A == B is False

The Given A!= B is True

The Given A >= B is True

The Given A <= B is False

Output3:

Enter the A value: 15

Enter the B value: 15

Addition of a and b 30

Subtraction of a and b 0

Multiplication of a and b 225

Power of a and b 437893890380859375

Remainder of a and b 0

Quotient of a and b 1

Division of a and b 1.0

The given A > B is False

The given A < B is False

The given A == B is True

The Given A! = B is False

The Given A >= B is True

The Given A <= B is True

2. Prog to find which is big in given three numbers
   Aim: To find the biggest number in given three numbers using decision control statements in python.

   Algorithm:
   Step 1: Enter First number and read it into A.
   Step 2: Enter First number and read it into B.
   Step 3: Enter First number and read it into C.
   Step 4: Using If condition statement writes the following code.

```
if a>b and a>c:
        print(a,"Is The big in given numbers")
elif b>a and b>c:
        print(b,"Is The big in given numbers")
else:
        print(c, "Is The big in given numbers ")
```

   Step 5: Stop.

Program:

```
# Program to print the big number in given three numbers.

print(" This is a Program to display big in given three numbers in Python")
a=int(input("Enter A Value: "))
b=int(input("Enter B Value: "))
c=int(input("Enter C Value: "))
if a>b and a>c:
   print(" The given A value",a,"Is The big in given numbers")
elif b>a and b>c:
   print("The given B value",b,"Is The big in given numbers")
else:
   print("The given C value",c,"Is The big in given numbers")
```

Output 1:

```
This is a Program to display big in given three numbers in Python
Enter A Value: 85
Enter B Value: 62
Enter C Value: 78
 The given A value 85 Is The big in given numbers
```

Output 2:

```
This is a Program to display big in given three numbers in Python
Enter A Value: 32
Enter B Value: 56
Enter C Value: 45
The given B value 56 Is The big in given numbers
```

3. Prog to print the multiplication table of given number.

Aim :-To display the multiplication table for a given number in Python.

Algorithm :-

Step1 : Start

Step 2 : Read the value of a number into ‘num’ to which the multiplication table is to be printed

Step 3 : Perform loop for ‘i’ in range(1,11) do the following:

1.Display(num,”X”,i,”=”,num*i)

Step 4 : Display the result

Step 5 : Stop

Program:

```
#program for to print multiplication of given number

num = int(input("Enter the number: "))
print("Multiplication Table of", num)
for i in range(1, 11):
   print(num,"X",i,"=",num * i)
```

Output:

```
Enter the number: 25
Multiplication Table of 25
25 X 1 = 25
25 X 2 = 50
25 X 3 = 75
25 X 4 = 100
25 X 5 = 125
25 X 6 = 150
25 X 7 = 175
25 X 8 = 200
25 X 9 = 225
25 X 10 = 250
```

4. Python program to find the factorial of a given number

Aim :-To find the factorial of a given number in Python.

Algorithm :-

Step1 : Start

Step 2 : Read the value of a number into ‘num’ to find its factorial value.

Step 3 : Assign factorial =1

Step 4 : Check if num < 0 then

    1. Display “Sorry, factorial does not exist for negative numbers “

Step 5 : Else check if num=0 then

    1. Display “The factorial of 0 is 1”

Step 6 : Else do the following :

    1. Perform loop for ‘i’ in range(1,num+1) do the following:
        1. Factorial=factorial *i

Step 7 : Display “The factorial value”,factorial

Step 8 : Stop

Program:

```
# Python program to find the factorial of a given number
num = int(input("Enter a number: "))
factorial = 1
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   for i in range(1,num + 1):
       factorial = factorial*i
   print("The factorial of",num,"is",factorial)
```

OUTPUT :

```
Enter a number: 8
The factorial of 8 is 40320
```

5. Python program to Create a list and perform the following methods
1) insert() 2) remove() 3) append() 4) len() 5) pop() 6) clear()

Aim :-To create a list and perform its operations in Python.

Algorithm :-

Step1 : Start

Step 2 : Create a list ‘a’ and assign values into the list

Step 3 : Display the values in the list ‘a’

Step 4 : Perform list operations like insert, remove, append, len, pop, clear, etc.

Step 5 : Display the values in the list ‘a’

Step 6 : Stop.

Program:

```
# program to Demonstrate Creating List and perform different operations on it.

a=[1,3,5,6,7,[3,4,5],"hello"]
print(a)
a.insert(3,20)
print(a)
a.remove(7)
print(a)
a.append("hi")
print(a)
len(a)
print(a)
a.pop()
print(a)
a.pop(6)
print(a)
a.clear()
print(a)
```

OUTPUT :

[1, 3, 5, 6, 7, [3, 4, 5], 'hello']

[1, 3, 5, 20, 6, 7, [3, 4, 5], 'hello']

[1, 3, 5, 20, 6, [3, 4, 5], 'hello']

[1, 3, 5, 20, 6, [3, 4, 5], 'hello', 'hi']

[1, 3, 5, 20, 6, [3, 4, 5], 'hello', 'hi']

[1, 3, 5, 20, 6, [3, 4, 5], 'hello']

[1, 3, 5, 20, 6, [3, 4, 5]]

[]

6. Write a program to define a module to find Fibonacci numbers and import the module in another program.

   Aim:- Program to demonstrate of defining a module and use in another program with example of finding Fibonacci series.

   Algorithm:

   <u>Program1:</u>

Step 1: Start.

Step 2: Defining first module with name Fibonacci.

Step 3: in that create user defined function using def <file_name>

Step 4: Read the values n1=0 and n2=1 and print n1 and n2

Step 5: and run the loop and print the value of n3 using following code

```
for x in range(0,n):
    n3=n1+n2
    if(n3>=n):
        break;
    print(n3,end=' ')
    n1=n2
    n2=n3
```

Step 6: Stop

Program2:

Step 1: Start.

Step 2: import the Fibonacci module.

Step 3: Enter a Value of the print range and Read the value into n

Step 4: Use the following code to call function in Fibonacci module

```
if(n<0):
    print("enter correct range!!")
else:
    print("------------------------FIBONACCI SERIES ---------------------- \n")
    mul.fibonacci(n)
```

Step 5: Stop

Program1:

```
def fibonacci(n):
    n1=0
    n2=1
    print(n1)
    print(n2)
    for x in range(0,n):
        n3=n1+n2
        if(n3>=n):
            break;
        print(n3,end=' ')
```

```
            n1=n2
            n2=n3
```

using_fibonacci.py

(Note: we will be using previous program as a library or package It is mandatory to write both the programs are separately)

Program2:

```
import fibonacci
n=int(input("enter range:"))
if(n<0):
print("enter correct range!!")
else:
print("-----------------------------FIBONACCI SERIES ----------------------------- \n")
fibonacci.fibonacci (n)
```

Output:

```
enter range:35

-----------------------------FIBONACCI SERIES -----------------------------


0

1

1 2 3 5 8 13 21 34
```

7. Program to create a file and perform different operations on it in python.

Aim: - To write a program create a file and perform different operations using file functions in python.

Procedure/Algorithm:
Step 1: Start.
Step 2: Create a file using file open command using write mode.
Step 3: Write mode write some text on it using write command.
Step 4: Read and print the content in the file.
Step 5: Close the file.
Step 6: Open the file again in the append mode.
Step 7: Add additional text in to file.
Step 8: Again, Read and print the content in the file.
Step 9: Close the file.
Step10: Stop.

Program:

```
f=open('myfile.txt','w')                    # Creating a file in write mode
f.write("Hi..Iam Creating a file First time")          # Write a Text into file
f=open('myfile.txt','r')                # Open a file in a read mode
print("****Content in the File Before Append additional Text add to it*****")
print(f.read()) # Read the content in File
f.close() # Closing the file.
f=open('myfile.txt','a')       # Open a file in append mode
f.write('\n Welcome Every one and  I am adding the additional text')
f.write('\n Append Mode')
f.close()

f=open('myfile.txt','r')       # Read the content in File after appending text
print("****Cotent in the File After Append additional Text add to it*****")
print(f.read())
f.close()
```

Output:

```
****Cotent in the File Before Append adittional Text add to it*****
Hi..Iam Creating a file First time
****Cotent in the File After Append adittional Text add to it*****
Hi..Iam Creating a file First time
 Welcome Every one and iam adding the additional text
 Append Mode
```

8). Python program to handle Division by Zero error using exception and check else and finally blocks.

Aim :-To handle division by zero error using exception in Python and Checking the how works else and finally Block.

Algorithm :-

Step1 : Start

Step 2 : Use 'try' block and do the following:

1. Read the values into two numbers 'a' and 'b'
2. Perform division c=a/b
3. Display the result 'c'

Step 3 : Use exception object 'e' to handle the division by zero error and do the following:

1. Display 'can't divide by zero'
2. Display exception object 'e'

Step 4 : Perform all relational operations on 'X' and 'Y'

Step 5 : Stop

Program:

```
try:
   a=int(input("Entera Value of A:"))
   b=int(input("Entera Value of B:"))
   c=a/b
   print("After dividing the Value, if not b value is 0 then:  ",c)
except:
   print("Any number Can not Divide by Zero")
else:
   print("This is else ")
finally:
   print("This exectues always:")
   x=5
   print("X value is given:", a)
   y=8
   print("Y value is given:", b)
   print(" After adding", x,"and", y,"is:",x+y)
```

Output1 :

```
Entera Value of A:12
Entera Value of B:6
After dividing the Value, if not b value is 0 then:   2.0
This is else
This exectues always:
X value is given: 12
Y value is given: 6
 After adding 5 and 8 is: 13
```

Output: 2

Entera Value of A:6
Entera Value of B:0
Any number Can not Divide by Zero
This exectues always:
X value is given: 6
Y value is given: 0
After adding 5 and 8 is: 13

9). Write a Program Demonstration of Packages in Python

Aim :- To Write a program create packages and use that in other programs in python

Algorithm :-

Step1 : Start
Step 2 : Create a new folder and named like 'mypackage' (D:\mypackage)
Step 3 : Using Python IDLE, Create a file or module with name 'greet.py' inside my package
Step 4 : Define function 'SayHello' in that file
Step 5 : Also, Create another file with name 'functions.py' inside my package
Step 6 : Define functions such as sum, average, power in that file
Step 7 : Import package 'mypackage' and call functions
Step 8 : Display the result
Step 9 : Stop

Program 1:

greet.py

```
#greet.py
def SayHello(name):
   print("Hello Hi this Suneel", name)
```

program 2:

functions.py

```
#functions.py
def sum(x,y):
   print("The sum of Given Two numbers are: ", x+y)
def average(x,y):
   print("The Average of Given Two numbers are: ", (x+y)/2)
def power(x,y):
   print("The power of Given Two numbers are: ", x**y)
```

program 3:

```
from mypackages import functions # importing functions module from mypackages
from mypackages import greet     # importing greet module from mypackages

a=int(input("Enter First value: "))
b=int(input("Enter First value: "))
text=" This demonstration of how package working"

greet.SayHello(text) # Calling SayHello Function in Greet module

functions.power(a,b) # Calling power Function in Greet module
functions.sum(a,b)
functions.average(a,b)
```

Output:

```
Enter First value: 25
Enter First value: 3
Hello Hi this Suneel This demonstration of how package working
The power of Given Two numbers are:  15625
The sum of Given Two numbers are:  28
The Average of Given Two numbers are:  14.0
```

10. Program to demonstration of multi-threading concept using python.

Aim: To write a program to know the how the cocept of multithreading working using python.

Procedure/ Algorithm:

Step 1: Start.
Step 2: Import predefined threading and time modules.
Step 3: Define the functions with name cal_square and cal_qube using following code.

```
            def cal_square(num):
        print("Calculate the Square of given Number ")
        for n in num:
            time.sleep(0.4)
            print(" Square of the given number is :", n*n)

    def cal_qube(num):
        print(" Calculate the Qube of given Number")
        for n in num:
            time.sleep(0.5)
            print("Qube of the given number is :", n*n*n)
```

Step 4: Create thread using threading. Thread(target=<function nae>, args=(arguments,)).
Step 5: Start the thread using thread start method. And also use the thread join method for stopping the next line of instruction executions stops until the completion of thread execution.
Step 6: Observe the output and time completion of the threads.
Steps 7: Stop.

Program:

```
import threading
import time

def cal_square(num):
    print("Calculate the Square of given Number ")
    for n in num:
        time.sleep(0.4)
        print(" Square of the given number is :", n*n)

def cal_qube(num):
    print(" Calculate the Qube of given Number")
    for n in num:
        time.sleep(0.5)
        print("Qube of the given number is :", n*n*n)


t2=time.time()

print(t2)
li=[5,6,8,9,3]
s1=threading.Thread(target=cal_square, args=(li, )) # define Thread1
s2=threading.Thread(target=cal_qube,args=(li, ))     # define Thread2
s1.start()  # Starting Thread1
s2.start()   # Starting Thread2
s1.join() # Stoping After Thread1,line Execution Untill completion of thread1
s2.join()  # Stoping After Thread2, line Execution Untill completion of thread2


print(" Total Time taken by Threadis:",time.time()-t2)
```

Output:

```
1664086489.294751
Calculate the Square of given Number  Calculate the Qube of given Number

 Square of the given number is : 25
Qube of the given number is : 125
```

```
 Square of the given number is : 36
Qube of the given number is : 216
 Square of the given number is : 64
Qube of the given number is : 512
 Square of the given number is : 81
Qube of the given number is : 729
 Square of the given number is : 9
Qube of the given number is : 27
 Total Time taken by Threadis: 2.8329215049743652
```

11. Program To display Digital Clock using Tkinter in Python

Aim: To write a program to create a digital clock using Tkinter in python

Procedure/ Algorithm:-

Step 1: Start.
Step 2: Import the all modules from Tkinter package.
Step 3 : Set the color and size of window to display the clock using geometry and config functons.
Step 4: Create one function to window using following code to display the time format.

```
def update():
   clock.config(text=time.strftime("%I:%M:%S"))
   clock.after(1000,update)
```

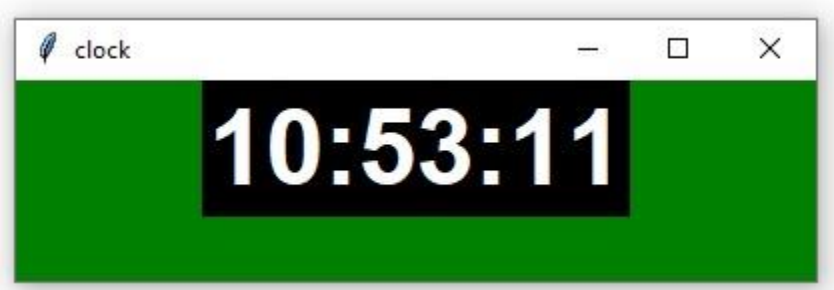
Step 5: Add label to window to window to display the style, color and size of the font.
Step 6: Call the pack() method and update() to display the clock.
Step 7: Run the program and Observe the output.
Step 8: Stop.

Program:

```
from tkinter import *
import time

root= Tk()
root.geometry("400x100")
root.config(bg='green')

def update():
   clock.config(text=time.strftime("%I:%M:%S"))
   clock.after(1000,update)


clock = Label(root, background = 'black',foreground = 'white', font = ('arial', 40, 'bold'))
clock.pack()
```

```
update()
root.title('clock')
root.mainloop()
```

Output:



12. Program to create a database and insert records into database and display the records using Python.

Aim: To write a program to create a data base, inserting records into data base and display the records in it.

Procedure/ Algorithm.

Step 1: Start.
Step 2: import the sqlite package for connecting the database.
Step 3: using sqlite.connect(‘dbname.db’) create a database.
Step 4: using sqllite.execute(“Sql query for Creating table/Inserting Records/Display Records”), we create a table in Database and insert the records into that after inserting the records display the data in that.
Step 5: Run the program and observe the out put.
Step 6:stop.

Program:

```
import sqlite3

conn = sqlite3.connect('test.db')
print("Opened database successfully")

conn.execute('''CREATE TABLE EMPLOYEE
       (ID INT PRIMARY KEY     NOT NULL,
       NAME           TEXT    NOT NULL,
       AGE            INT     NOT NULL,
       ADDRESS        CHAR(50),
       SALARY         REAL);''')
print ("Table created successfully")
```

```
print ("Now I Enter Records in to Database")

conn.execute("INSERT INTO EMPLOYEE (ID,NAME,AGE,ADDRESS,SALARY) \
     VALUES (10, 'SUNEEL', 37, 'Nellore', 20000.00 )");

conn.execute("INSERT INTO EMPLOYEE (ID,NAME,AGE,ADDRESS,SALARY) \
     VALUES (20, 'SRI', 29, 'Tirupathi', 15000.00 )");

conn.execute("INSERT INTO EMPLOYEE (ID,NAME,AGE,ADDRESS,SALARY) \
     VALUES (30, 'Reddy', 23, 'Nellore', 20000.00 )");

conn.execute("INSERT INTO EMPLOYEE (ID,NAME,AGE,ADDRESS,SALARY) \
     VALUES (40, 'Mars', 25, 'Chittur ', 65000.00 )");

conn.commit()
print ("Records created successfully")
print ("And Display records in database successfully");

cursor = conn.execute("SELECT id, name, address, salary from EMPLOYEE ")
for row in cursor:
   print( "ID = ", row[0])
   print ("NAME = ", row[1])
   print ("ADDRESS = ", row[2])
   print ("SALARY = ", row[3], "\n")

print ("Operation done successfully");
conn.close()
```

Output:

```
Opened database successfully
Table created successfully
Now I Enter Records in to Database
Records created successfully
And Display records in database successfully
ID =  10
NAME =  SUNEEL
ADDRESS =  Nellore
SALARY =  20000.0

ID =  20
NAME =  SRI
ADDRESS =  Tirupathi
SALARY =  15000.0

ID =  30
```

NAME = Reddy
ADDRESS = Nellore
SALARY = 20000.0

ID = 40
NAME = Mars
ADDRESS = Chittur
SALARY = 65000.0

Operation done successfully