

DEPARTMENT OF COMPUTER SCIENCE

RECORD NOTE

Record work submitted to the Bharathiar University in partial fulfillment of the
requirement for the Degree of

Master of Science in Computer Science



PROGRAMMING LAB - Python Programming Lab

SHRI NEHRU MAHA VIDYALAYA COLLEGE OF ARTS & SCIENCE

(Affiliated to Bharathiar University)

Shri Gambhirmal Bafna Nagar

Malumachampatti, Coimbatore-641050

OCTOBER – 2024



SHRI NEHRU MAHA VIDYALAYA

COLLEGE OF ARTS AND SCIENCE

SNMV

(Affiliated to Bharathiar University, Coimbatore, Re-accredited with "A" Grade by NAAC)
Shri Gambhirmal Bafna Nagar, Malumachampatti, Coimbatore - 641 050, Tamil Nadu, India.



MASTER OF SCIENCE IN COMPUTER SCIENCE

This is to certify that it is a bonafide record work done by

_____ Studying I year M.Sc . Computer Science

Reg .No. _____

Staff In – Charge

Head of the Department

Submitted for: Practical II : Python Programming Lab

Bharathiar University Practical Examination held on _____

Odd semester (2024-2025)

CONTENT

S.NO	DATE	TITLE	PAGE NO.
1.		Programs using Elementary data items Lists, Dictionaries and Tuples	
2.		Program using Conditional Branches	
3.		Programs using loops	
4.		Programs using functions	
5.		Programs using Exception Handling	
6.		Programs using Inheritance	
7.		Programs using Polymorphism	
8.		Programs to implement file operations	
9.		Programs using Modules	
10.		Programs for creating Dynamic and Interactive web pages using forms	

Staff In-Charge

EX.NO: 1	ELEMENTARY DATA ITEMS LISTS, DICTIONARIES AND TUPLES
DATE:	

AIM:

To implement a python program using elementary data items, lists, dictionaries and tuples.

ALGORITHM:

Step 1: Start the program .

Step2 : Write a code for creating python list .

Step 3: Write a code creating python strings .

Step 4: Write a code creating python tuples .

Step 5: Write a code creating python Dictionary .

Step 6: Save and run the program .

Step 7: Output will be displayed .

Step 8: Stop the program .

PROGRAM:

```
metros = ['mumbai','chennai','delhi']
```

```
print('List output')
```

```
for metro in metros:
```

```
    print(metro)
```

```
print()
```

```
print('Tuple output')
```

```
courses = ('bsc','bcom','msc')
```

```
print(courses)
```

```
print()
```

```
print('Dictionary output')
```

```
navisys = {'india':'navic','usa':'gps','russia':'glonass'}
```

```
print(navisys['india'])
```

OUTPUT:

```
List output
mumbai
chennai
delhi

Tuple output
('bsc', 'bcom', 'msc')

Dictionary output
navic

...Program finished with exit code 0
Press ENTER to exit console.
```

RESULT:

Thus the above program was executed and the output was verified successfully.

EX.NO: 2	CONDITIONAL BRANCHES
DATE:	

AIM:

To implement a python program using Conditional branches.

ALGORITHM:

Step 1 : Start the Program .

Step 2 : Include the required variables .

Step 3 : Use conditional statements like if and else to print the millenium year of birth .

Step 4 : Save and run the program .

Step 5 : Output will be displayed .

Step : Stop the program .

PROGRAM:

```
yearofbirth= 1985
```

```
if yearofbirth >= 1 and yearofbirth <= 1000:
```

```
    print('Born in First Millenium')
```

```
elif yearofbirth>1000 and yearofbirth <= 2000:
```

```
    print('Born in Second Millenium')
```

```
elif yearofbirth>2000 and yearofbirth <= 3000:
```

```
    print('Born in Third Millenium')
```

```
else:
```

```
    print('Please enter the BirthYear between 1 to 3000')
```


OUTPUT:

```
Born in Second Millenium
```

```
...Program finished with exit code 0  
Press ENTER to exit console.[]
```

RESULT :

Thus the above program was executed and the output was verified successfully.

EX.NO: 3	LOOPS
DATE:	

AIM:

To implement a Python program using for and while loops.

ALGORITHM:

Step 1: Start the program ..

Step 2: Write a code for illustrate the loops .

Step 3: Write a code for illustrate while loops .

Step 4: Write a code for illustrate for loops.

Step 5: Save and run the program .

Step 6: Output will be displayed .

Step 7: Stop the program .

PROGRAM:

For loop:

```
Missiles = [ 'Prithvi','Agni','Nirbhay']
```

```
for m in Missiles:  
    print(m,end=' ')
```

While loop:

```
num=0
```

```
sum=0
```

```
while num<=10:
```

```
    sum = sum + num
```

```
    num += 2
```

```
print(sum)
```

OUTPUT:

For loop:

```
Prithvi Agni Nirbhay  
...Program finished with exit code 0  
Press ENTER to exit console.
```

While loop:

```
30  
...Program finished with exit code 0  
Press ENTER to exit console.
```

RESULT:

Thus the above program was executed and the output was verified successfully.

EX.NO: 4	FUNCTIONS
DATE:	

AIM:

To implement a python program using functions .

ALGORITHM:

Step 1: Start the program .

Step 2: Write a code for using functions to find out the leap year.

Step 3: Save the program and run the program .

Step 4: Output will be displayed .

Step 5: Stop the program .

PROGRAM:

```
def findleap(year1):  
    if(year1%4 == 0):  
        print(year1,'is a leap year')  
    else:  
        print(year1,'is not a leap year')  
  
year=int(input("Enter the year "))  
findleap(year)
```

OUTPUT:

```
Enter the year 2020
2020 is a leap year

...Program finished with exit code 0
Press ENTER to exit console.[]
```

RESULT :

Thus the above program was executed and the output was verified successfully.

EX.NO: 5	EXCEPTION HANDLING
DATE:	

AIM:

To implement a python program to illustrate the use of Exceptions .

ALGORITHM:

Step 1 : Start the program .

Step 2 : Declare the required variables .

Step 3 : Include the necessary exceptions like try and except .

Step 4 : Save and run the program .

Step 5 : Output will be displayed .

Step 6 : Stop the program .

PROGRAM:

```
class invalidsalary(Exception):  
  
    pass  
  
salary=1  
  
try:  
    input_sal = int(input("Enter salary: "))  
  
    if input_sal < salary:  
  
        raise invalidsalary  
  
    else:  
  
        print("Let's hope for a good increment")  
  
except invalidsalary:  
  
    print("Salary cannot be zero (or) -ve value")
```

OUTPUT:

```
Enter salary: -234
Salary cannot be zero (or) -ve value

...Program finished with exit code 0
Press ENTER to exit console.[]
```

```
Enter salary: 5000
Let's hope for a good increment

...Program finished with exit code 0
Press ENTER to exit console.[]
```

RESULT :

Thus the above program was executed and the output was verified successfully.

EX.NO: 6	INHERITANCE
DATE:	

AIM:

To implement a python program to illustrate the inheritance .

ALGORITHM:

Step 1: Start the program .

Step 2: Create a base class .

Step 3: Create a subclass which include the properties of the base class .

Step 4: Write a code for relevant class .

Step 5: Save and run the program .

Step 6: Output will be displayed .

Step 7: Stop the program .

PROGRAM:

```
class A:
```

```
    i=0
```

```
    j=0
```

```
class B(A):
```

```
    k=0
```

```
    def sum(self):
```

```
        i=int(input("Enter first number "))
```

```
        j=int(input("Enter second number "))
```

```
        k=int(input("Enter third number "))
```

```
        x=i+j+k
```

```
        print('The sum is ',x)
```

```
obj = B()
```

```
obj.sum()
```

OUTPUT:

```
Enter first number 45
Enter second number 56
Enter third number 67
The sum is 168

...Program finished with exit code 0
Press ENTER to exit console.
```

RESULT :

Thus the above program was executed and the output was verified successfully.

EX.NO: 7	POLYMORPHISM
DATE:	

AIM:

To implement a python program to illustrate the polymorphism .

ALGORITHM:

Step 1 : Start the program .

Step 2 : Declare the required variables .

Step 3 : Add integer and float values to represent the polymorphism .

Step 4 : Save and run the program .

Step 5 : Output will be displayed .

Step 6 : Stop the program .

PROGRAM:

```
class intadd:
```

```
    def __init__(self, num1, num2):
```

```
        self.num1=num1
```

```
        self.num2=num2
```

```
    def sum(self):
```

```
        result=self.num1+self.num2
```

```
        print(result)
```

```
class floatadd:
```

```
    def __init__(self, no1, no2):
```

```
        self.no1=no1
```

```
        self.no2=no2
```

```
    def sum(self):
```

```
        result=self.no1+self.no2
```

```
        print(result)
```

```
ob1= intadd(18,23)
```

```
ob2=floatadd(7.2,6.8)
```

```
for k in(ob1,ob2):
```

```
    k.sum()
```

OUTPUT:

```
The integer addition 41  
The float addition 14.0
```

RESULT:

Thus the above program was executed and the output was verified successfully.

EX.NO: 8	IMPLEMENTING FILE OPERATIONS
DATE:	

AIM:

To implement a python program to illustrate the file operations .

ALGORITHM:

Step 1: Start the program .

Step 2: create a function called read(), to read the contents of the file .

Step 3: create a function called open(), to opening the contents of the file .

Step 4: create a function called write(), to write the contents to the file .

Step 5 : Save and run the program .

Step 6 : Output will be displayed .

Step 7 : Stop the program .

PROGRAM:

```
# open a file in read mode
file1 = open("file1.txt")

# read the file content
read_content = file1.read()
print(read_content)
```

Output:



```
python files
very good
```

Writing into the file

Program :

```
file = open('file.txt','w')
file.write("Welcome to CS ")
file.write("M.SC(CS)")
file.close()
```

Output :



```
Welcome to CS M.SC(CS)
```

create file1.txt in notepad in the same directory
type some contents in the file
execute the above program
verify the output

Appending into the file

Program:

```
file = open('file.txt','a')
file.write(" jai bharath ")
file.write("bharath matha ki jai")
file.close()
```

Output:

```
Welcome to CS M.SC(CS) jai bharath bharath matha ki jai
```

Renaming the file

Program:

```
import os

#rename file.txt to file9.txt
os.rename("file.txt","file9.txt")

check the name of the file in directory
```

Creating directory

Program:

```
import os

#creating a new directory with the name new
os.mkdir("CS")

check the directory for output
```

Removing the file

Program:

```
import os;
#deleting the file named file3.txt
os.remove("file1.txt")

check the directory for output
```

RESULT:

Thus the above program was executed and the output was verified successfully.

EX.NO: 9	MODULES
DATE:	

AIM:

To implement a python program to illustrate the modules .

ALGORITHM:

Step 1 : Start the program .

Step 2 : Declare the required variables .

Step 3 : With the use of modules to calculate the values of sum , average and power .

Step 4 : Save and run the program .

Step 5 : Output will be displayed .

Step 6 : Stop the program .

PROGRAM:

Save the following program as mymodule.py

```
def sum(x,y):  
  
    return x+y  
  
def average(x,y):  
  
    return (x+y)/2  
  
def power(x,y):  
  
    return x**y
```

Save the following program as modtest.py

```
import mymodule  
  
print ("sum:",mymodule.sum(10,20))  
  
print("Average:",mymodule.average(10,20))  
  
print("Power:",mymodule.power(10,2))
```

OUTPUT:

```
sum: 30
```

```
Average: 15.0
```

```
Power: 100
```

RESULT :

Thus the above program was executed and the output was verified successfully.

EX.NO: 10	WEB PAGES
DATE:	

AIM:

To implement a python program for creating Dynamic and Interactive web pages using forms .

ALGORITHM:

Step 1: Start Pycharm IDE

Step 2: Start the XAMPP server for program execution from the XAMPP control panel

Step 3: Type the following command in pycharm terminal window to import mysql connector
pip install mysql-connector-python

Step 4: Create a new project in pycharm and a new file for database connectivity program

Step 5: Type the coding

Step 6: Create new database and table in phpMyAdmin

Step 7: Insert rows into the table

Step 8: Execute the program in pycharm, the rows available in the table will be displayed in the output

PROGRAM:

```
import mysql.connector
from mysql.connector import Error

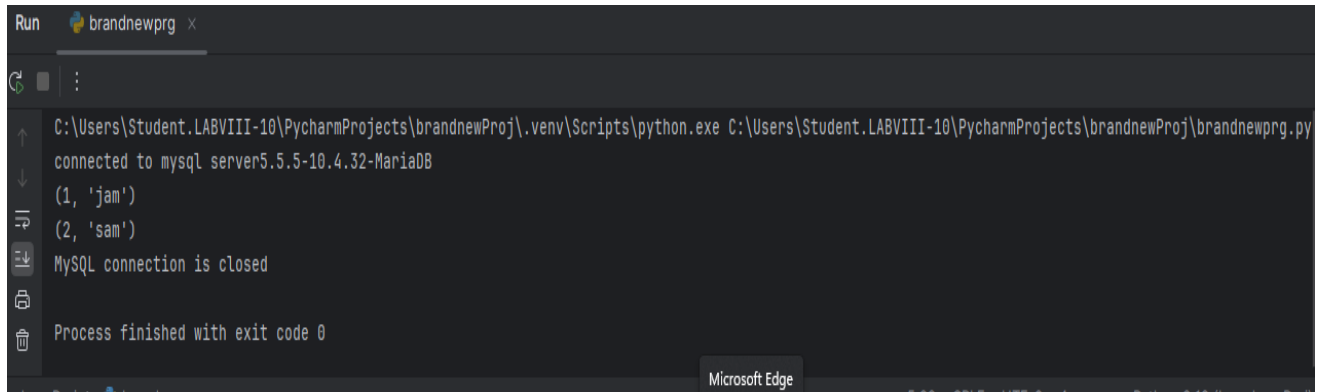
try:
    conn = mysql.connector.connect(host = 'localhost',database = 'labdb',user =
'root',password = "")
    if conn.is_connected():
        db = conn.get_server_info()
        print("connected to mysql server " + db)
        cursor = conn.cursor()
        cursor.execute('select * from student')
        record = cursor.fetchall()
        for result in record:
            print(result)

except Error as e:
    print("error while connecting database",e)
finally:
    if conn.is_connected():
        cursor.close()
        conn.close()
        print("connection closed")
```

type the following in terminal

```
pip install mysql-connector-python
```


OUTPUT:



```
Run brandnewprg x
C:\Users\Student.LABVIII-10\PycharmProjects\brandnewProj\.venv\Scripts\python.exe C:\Users\Student.LABVIII-10\PycharmProjects\brandnewProj\brandnewprg.py
connected to mysql server5.5-10.4.32-MariaDB
(1, 'jam')
(2, 'sam')
MySQL connection is closed
Process finished with exit code 0
```

RESULT:

Thus the above program was executed and the output was verified successfully.