

COMPUTER SCIENCE

PRACTICAL PROGRAMS WITH SOLUTION

Practical Hand Book

General Instructions:

1. Eight Exercises from Python and Two from MySQL are practiced in the practical classes.
2. In Practical exams, the question paper will have two questions with internal choice.
3. One question should be chosen from the list of internal choice.
4. Distribution of Marks as follows:

Duration of Practical: 2 ½ Hrs

Maximum Marks: 20

I. Internal Assessment: 5 Marks

Record Book 5 Marks

II. External Assessment: 15 Marks

Writing Code 10 Marks

Execution 5 Marks

Total	20 Marks
--------------	-----------------

INDEX

Sl. No.	Question Number	Program Name	Page Number
1	PY1	(a) Calculate Factorial (b) Sum of Series	3
2	PY2	(a) Odd or Even (b) Reverse the String	4
3	PY3	Generate values and remove odd numbers	5
4	PY4	Generate Prime numbers and Set Operations	6
5	PY5	Display a String elements – Using Class	7
6	DB6	MySQL – Employee Table	9
7	DB7	MySQL – Student Table	12
8	PY8	Python with CSV	16
9	PY9	Python with SQL	18
10	PY10	Python Graphics with Pip	20

▮ PY1(a) - Calculate Factorial ▮

1(a) Write a program to calculate the factorial of the given number using for loop

Coding

```
num = int(input("Enter a Number: "))
if (num==0):
    fact = 1
fact = 1
for i in range(1,num+1):
    fact = fact * i
print("Factorial of ", num, " is ", fact)
```

▮ Output: ▮

```
Enter a Number: 12
Factorial of 12 is 479001600
```

▮ PY1(b) - Sum of Series ▮

1(b) Write a program to sum the series: $1/1 + 2^2/2 + 3^3/3 + \dots + n^n/n$

Coding

```
n = int(input("Enter a value of n: "))
s=0.0
for i in range(1,n+1):
    a=float(i**i)/i
    s=s+a
print("The sum of the series is ", s)
```

▮ Output: ▮

```
Enter a value of n: 4
The sum of the series is 76.0
```

PY2(a) - Odd or Even ↩

2(a) Write a program using functions to check whether a number is even or odd

Coding

```
def oddeven(a):
    if (a%2==0):
        return 1
    else:
        return 0
num = int(input("Enter a number: "))
if (oddeven(num)==1):
    print("The given number is Even")
elif (oddeven(num)==0):
    print("The given number is Odd")
```

Output: ↩

```
Enter a number: 7
The given number is Odd
Enter a number: 6
The given number is Even
```

PY2(b) - Reverse the String ↩

2(b) Write a program to create a mirror of the given string. For example, "wel" = "lew".

Coding

```
def rev(str1):
    str2=""
    i=len(str1)-1
    while i>=0:
        str2+=str1[i]
        i-=1
    return str2
word = input("\n Enter a String: ")
print("\n The Mirror image of the given string is: ", rev(word))
```

Output: ↩

```
Enter a String: school
The Mirror image of the given string is: loohcs
```

PY3 – Generate values and remove odd numbers ↩

3(a)

Write a program to generate values from 1 to 10 and then remove all the odd numbers from the list

Coding

```
num1=[]  
for i in range(1,11):  
    num1.append(i)  
print("Numbers from 1 to 10.....\n",num1)  
  
for j, i in enumerate(num1):  
    if(i%2==1):  
        del num1[j]  
print("The values after removed odd numbers.....\n",num1)
```

Output: ↩

Numbers from 1 to 10.....

[10 ,9 ,8 ,7 ,6 ,5 ,4 ,3 ,2 ,1]

The values after removed odd numbers.....

[10 ,8 ,6 ,4 ,2]

PY4 – Generate Prime numbers and Set Operations

4.

Write a Program that generate a set of prime numbers and another set of odd numbers. Display the result of union, intersection, difference and symmetric difference operations

Coding

```
odd=set([x*1+2 for x in range(0,5)])
primes=set()
for i in range(2,10):
    j=2
    f=0
    while j<i/2:
        if i%j==0:
            f=1
            j+=1
    if f==0:
        primes.add(i)
print("Odd Numbers: ", odd)
print("Prime Numbers: ", primes)
print("Union: ", odd.union(primes))
print("Intersection: ", odd.intersection(primes))
print("Difference: ", odd.difference(primes))
print("Symmetric Difference: ", odd.symmetric_difference(primes))
```

Output: ↪

Odd Numbers: {9 ,7 ,5 ,3 ,1}

Prime Numbers: {7 ,5 ,4 ,3 ,2}

Union: {9 ,7 ,5 ,4 ,3 ,2 ,1}

Intersection: {7 ,5 ,3}

Difference: {9 ,1}

Symmetric Difference: {9 ,4 ,2 ,1}

PY5 – Display sting elements – Using Class

5.

Write a program to accept a string and print the number of uppercase, lowercase, vowels, consonants and spaces in the given string using Class

Coding

```
class String:
    def __init__(self):
        self.uppercase=0
        self.lowercase=0
        self.vowels=0
        self.consonants=0
        self.spaces=0
        self.string=""

    def getstr(self):
        self.string=str(input("Enter a String: "))

    def count_upper(self):
        for ch in self.string:
            if (ch.isupper()):
                self.uppercase+=1

    def count_lower(self):
        for ch in self.string:
            if (ch.islower()):
                self.lowercase+=1

    def count_vowels(self):
        for ch in self.string:
            if (ch in ('A', 'a', 'e', 'E', 'i', 'I', 'o', 'O', 'l', 'L')):
                self.vowels+=1

    def count_consonants(self):
        for ch in self.string:
            if (ch not in ('A', 'a', 'e', 'E', 'i', 'I', 'o', 'O', 'l', 'L')):
                self.consonants+=1

    def count_space(self):
        for ch in self.string:
            if (ch==" "):
```

```
self.spaces+=1
```

```
def execute(self):
```

```
    self.count_upper()
```

```
    self.count_lower()
```

```
    self.count_vowels()
```

```
    self.count_consonants()
```

```
    self.count_space()
```

```
def display(self):
```

```
    print("The given string contains...")
```

```
    print("%d Uppercase letters"%self.uppercase)
```

```
    print("%d Lowercase letters"%self.lowercase)
```

```
    print("%d Vowels"%self.vowels)
```

```
    print("%d Consonants"%self.consonants)
```

```
print("%d Spaces"%self.spaces)
```

```
S = String()
```

```
S.getstr()
```

```
S.execute()
```

```
S.display()
```

Output:

Enter a String: Welcome to Computer Science

The given string contains...

3 Uppercase letters

21 Lowercase letters

10 Vowels

17 Consonants

3 Spaces

DB6 – MySQL Employee Table ↵

6.

Create an Employee Table with the fields Empno, Empname, Desig, Dept, Age and Place. Enter five records into the table

- Add two more records to the table.
- Modify the table structure by adding one more field namely date of joining.
- Check for Null value in doj of any record.
- List the employees who joined after 2018/01/01.

SQL Queries and Output:

(i) Creating Table Employee

```
mysql> Create table Employee (Empno integer(4) primary key,
Empname varchar(20), Desig varchar(10), Dept varchar(10),
Age integer(2), Place varchar(10));
```

(ii) View Table Structure:

```
mysql> Desc Employee;
```

Field	Type	Null	Key	Default	Extra
Empno	int(4)	NO	PRI	NULL	
Empname	varchar(20)	YES		NULL	
Desig	varchar(10)	YES		NULL	
Dept	varchar(10)	YES		NULL	
Age	int(2)	YES		NULL	
Place	varchar(10)	YES		NULL	

6 rows in set (0.00 sec)

(iii) Inserting Data into Table:

```
mysql> Insert into employee values(1221, 'Sidharth', 'Officer', 'Accounts', 45, 'Salem');
mysql> Insert into employee values(1222, 'Naveen', 'Manager', 'Admin', 32, 'Erode');
mysql> Insert into employee values(1223, 'Ramesh', 'Clerk', 'Accounts', 33, 'Ambathur');
mysql> Insert into employee values(1224, 'Abinaya', 'Manager', 'Admin', 28, 'Anna Nagar');
mysql> Insert into employee values(1225, 'Rahul', 'Officer', 'Accounts', 31, 'Anna Nagar');
```

(iv) Select all the record:

```
mysql> select * from Employee;
```

Empno	Empname	Desig	Dept	Age	Place
1221	Sidharth	Officer	Accounts	45	Salem
1222	Naveen	Manager	Admin	32	Erode
1223	Ramesh	Clerk	Accounts	33	Ambathur
1224	Abinaya	Manager	Admin	28	Anna Nagar
1225	Rahul	Officer	Accounts	31	Anna Nagar

5 rows in set (0.00 sec)

(v) Adding two more records:

```
mysql> Insert into employee values(3226, 'Sona', 'Manager', 'Accounts', 42, 'Erode');
```

```
mysql> Insert into employee values(3227, 'Rekha', 'Officer', 'Admin', 34, 'Salem');
```

```
mysql> select * from Employee;
```

Empno	Empname	Desig	Dept	Age	Place
1221	Sidharth	Officer	Accounts	45	Salem
1222	Naveen	Manager	Admin	32	Erode
1223	Ramesh	Clerk	Accounts	33	Ambathur
1224	Abinaya	Manager	Admin	28	Anna Nagar
1225	Rahul	Officer	Accounts	31	Anna Nagar
3226	Sona	Manager	Accounts	42	Erode
3227	Rekha	Officer	Admin	34	Salem

7 rows in set (0.00 sec)

(vi) Adding one more Field:

```
mysql> Alter table employee add(doj date);
```

```
desc employee;
```

Field	Type	Null	Key	Default	Extra
Empno	int(4)	NO	PRI	NULL	
Empname	varchar(20)	YES		NULL	
Desig	varchar(10)	YES		NULL	
Dept	varchar(10)	YES		NULL	
Age	int(2)	YES		NULL	

Place	vvarchar(10)	YES		NULL	
doj	date	YES		NULL	

7 rows in set (0.00 sec)

(vii) Inserting date of joining to each employee:

```
mysql> update employee set doj = '21-03-2010' where empno=1221;
mysql> update employee set doj = '13-05-2012' where empno=1222;
mysql> update employee set doj = '25-10-2017' where empno=1223;
mysql> update employee set doj = '17-06-2018' where empno=1224;
mysql> update employee set doj = '02-01-2018' where empno=1225;
mysql> update employee set doj = '31-12-2017' where empno=3226;
mysql> update employee set doj = '16-08-2015' where empno=3227;
```

```
mysql> select * from Employee;
```

Empno	Empname	Desig	Dept	Age	Place	doj
1221	Sidharth	Officer	Accounts	45	Salem	2010-03-21
1222	Naveen	Manager	Admin	32	Erode	2012-05-13
1223	Ramesh	Clerk	Accounts	33	Ambathur	2017-10-25
1224	Abinaya	Manager	Admin	28	Anna Nagar	2018-06-17
1225	Rahul	Officer	Accounts	31	Anna Nagar	2018-01-02
3226	Sona	Manager	Accounts	42	Erode	2017-12-31
3227	Rekha	Officer	Admin	34	Salem	2015-08-16

7 rows in set (0.00 sec)

(viii) Checking null value in doj

```
mysql> select * from emp where empno is null;
Empty set (0.00 sec)
```

(ix) List the employees who joined after 2018/01/01.

```
mysql> Select * from emp where doj > '01-01-2018';
```

Empno	Empname	Desig	Dept	Age	Place	doj
1224	Abinaya	Manager	Admin	28	Anna Nagar	2018-06-17
1225	Rahul	Officer	Accounts	31	Anna Nagar	2018-01-02

2 rows in set (0.00 sec)

DB7 – MySQL Student Table ↩

7

Create Student table with following fields and enter data as given in the table below

Field	Type	Size
Reg_No	char	5
Sname	varchar	15
Age	int	2
Dept	varchar	10
Class	char	3

Data to be entered

Reg_No	Sname	Age	Dept	Class
M1001	Harish	19	ME	ME1
M1002	Akash	20	ME	ME2
C1001	Sneha	20	CSE	CS1
C1002	Lithya	19	CSE	CS2
E1001	Ravi	20	ECE	EC1
E1002	Leena	21	EEE	EE1
E1003	Rose	20	ECE	EC2

Then, query the followings:

- (i) List the students whose department is “CSE”.
- (ii) List all the students of age 20 and more in ME department.
- (iii) List the students department wise.
- (iv) Modify the class M2 to M1.
- (v) Check for the uniqueness of Register no.

SQL Queries and Output:

(1) Creating Table - Student

```
mysql>Create table Student(Reg_Nochar(5), Sname varchar(20), Age integer(2),
Dept varchar(10), Class char(3));
```

Query OK, 0 rows affected (0.51 sec)

View table structure:

```
mysql> desc Student;
```

Field	Type	Null	Key	Default	Extra
Reg_No	char(5)	YES		NULL	
Sname	varchar(20)	YES		NULL	
Age	int(2)	YES		NULL	
Dept	varchar(10)	YES		NULL	
Class	char(3)	YES		NULL	

5 rows in set (0.02 sec)

(2) Inserting Data into table:

```
mysql>Insert into Student values ('M1001', 'Harish', 19, 'ME', 'ME1');
```

```
mysql>Insert into Student values ('M1002', 'Akash', 20, 'ME', 'ME2');
```

```
mysql>Insert into Student values ('C1001', 'Sneha', 20, 'CSE', 'CS1');
```

```
mysql>Insert into Student values ('C1002', 'Lithya', 19, 'CSE', 'CS2');
```

```
mysql>Insert into Student values ('E1001', 'Ravi', 20, 'ECE', 'EC1');
```

```
mysql>Insert into Student values ('E1002', 'Leena', 21, 'EEE', 'EE1');
```

```
mysql>Insert into Student values ('E1003', 'Rose', 20, 'ECE', 'EC2');
```

View all records:

```
mysql> select * from Student;
```

Reg_No	Sname	Age	Dept	Class
M1001	Harish	19	ME	ME1
M1002	Akash	20	ME	ME2
C1001	Sneha	20	CSE	CS1
C1002	Lithya	19	CSE	CS2

E1001	Ravi	20	ECE	EC1
E1002	Leena	21	EEE	EE1
E1003	Rose	20	ECE	EC2

7 rows in set (0.00 sec)

(3) Other Queries:

(i) List the students whose department is "CSE":

mysql> Select * from Student where Dept='CSE';

Reg_No	Sname	Age	Dept	Class
C1001	Sneha	20	CSE	CS1
C1002	Lithya	19	CSE	CS2

2 rows in set (0.03 sec)

(ii) List all the students of age 20 and more in ME department:

mysql> Select * from Student where Age >=20 and Dept='ME';

Reg_No	Sname	Age	Dept	Class
M1002	Akash	20	ME	ME2

1 row in set (0.02 sec)

(iii) List the students department wise:

mysql> Select * from Student Group by Dept Order by Sname;

Reg_No	Sname	Age	Dept	Class
M1001	Harish	19	ME	ME1
E1002	Leena	21	CSE	EE1
E1001	Ravi	20	ECE	EC1
C1001	Sneha	20	EEE	CS1

4 rows in set (0.00 sec)

(iv) Modify the class M2 to M1:

mysql> Update Student set Class='ME1' where Class='ME2';

Query OK, 1 row affected (0.11 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from Student;

Reg_No	Sname	Age	Dept	Class
M1001	Harish	19	ME	ME1
M1002	Akash	20	ME	ME2
C1001	Sneha	20	CSE	CS1
C1002	Lithya	19	CSE	CS2
E1001	Ravi	20	ECE	EC1
E1002	Leena	21	EEE	EE1
E1003	Rose	20	ECE	EC2

7 rows in set (0.00 sec)

(v) Check for the uniqueness of Register no.

mysql> Select Distinct Reg_No from Student;

Reg_No
M1001
M1002
C1001
C1002
E1001
E1002
E1003

7 rows in set (0.02 sec)

PY8 - Python with CSV ↩

8

Write a program using python to get 10 players name and their score. Write the input in a csv file. Accept a player name using python. Read the csv file to display the name and the score. If the player name is not found give an appropriate message

Coding

```
import csv
with open('c:\\pyprg\\player.csv','w') as f:
    w = csv.writer(f)
    n=1
    while (n<=10):
        name = input("Player Name?:" )
        score = int(input("Score: "))
        w.writerow([name,score])
        n+=1
print("Player File created")
f.close()
searchname=input("Enter the name to be searched ")
f=open('c:\\pyprg\\player.csv','r')
reader = csv.reader(f)
lst=[]
for row in reader:
    lst.append(row)
q=0
for row in lst:
    if searchname in row:
        print(row)
        q+=1
if(q==0):
    print("string not found")
f.close()
```


Output:

Player Name?:Rohit Sharma

Score: 264

Player Name?:VirenderSehwag

Score: 219

Player Name?:Sachin Tendulkar

Score: 200

Player Name?:Dhoni

Score: 190

Player Name?:Sachin Tendulkar

Score: 250

Player Name?:ViratKohli

Score: 148

Player Name?:Ganguly

Score: 158

Player Name?:KapilDev

Score: 175

Player Name?:Amarnath

Score: 148

Player Name?:SunilGavaskar

Score: 200

Player File created

Enter the name to be searched Sachin Tendulkar

['Sachin Tendulkar', '200']

['Sachin Tendulkar', '250']

PY9 - Python with SQL ↩

9

Create a sql table using python and accept 10 names and age .sort in descending order of age and display

Coding

```
import sqlite3

connection = sqlite3.connect("info.db")

cursor = connection.cursor()

#cursor.execute("DROP Table student")

cursor.execute("create table student(name, age)")

print("Enter 10 students names and their ages respectively:")

for i in range(10):

    who =[input("Enter Name:")]

    age =[int(input("Enter Age:"))]

    n =len(who)

    for i in range(n):

        cursor.execute("insert into student values (?, ?)", (who[i],age[i]))

cursor.execute("select * from student order by age desc")

print("Displaying All the Records From student Table in Descending order of age")

print (*cursor.fetchall(),sep='\n' )
```

Output:

Enter 10 students names and their ages respectively:

Enter Name:Annamalai

Enter Age:17

Enter Name:Aashik Mathew

Enter Age:23

Enter Name:Kumaran

Enter Age:30

Enter Name:Sivasakthiya

Enter Age:28

Enter Name:Leena

Enter Age:45

Enter Name:Meena

Enter Age:65

Enter Name:Kamalakaran

Enter Age:35

Enter Name:Sowmyaa

Enter Age:20

Enter Name:Ramaa

Enter Age:70

Enter Name:Melvin

Enter Age:35

Displaying All the Records From student Table in Descending order of age

('Ramaa', 70)

('Meena', 65)

('Leena', 45)

('Kamalakaran', 35)

('Melvin', 35)

('Kumaran', 30)

('Sivasakthiya', 28)

('Aashik Mathew', 23)

('Sowmyaa', 20)

('Annamalai', 17)

PY10 – Python Graphics with Pip

10

Write a program to get five marks using list and display the marks in pie chart

Coding

```
import matplotlib.pyplot as plt
marks=[]
i=0
subjects = ["Tamil", "English", "Maths", "Science", "Social"]
while i<5:
    marks.append(int(input("Enter Mark = ")))
    i+=1
for j in range(len(marks)):
    print("{} Mark = {}".format(j+1, subjects[j],marks[j]))

plt.pie (marks, labels = subjects, autopct = "%.2f ")
plt.axes().set_aspect ("equal")
plt.show()
```

Output:

Enter Mark = 67

Enter Mark = 31

Enter Mark = 45

Enter Mark = 89

Enter Mark = 73

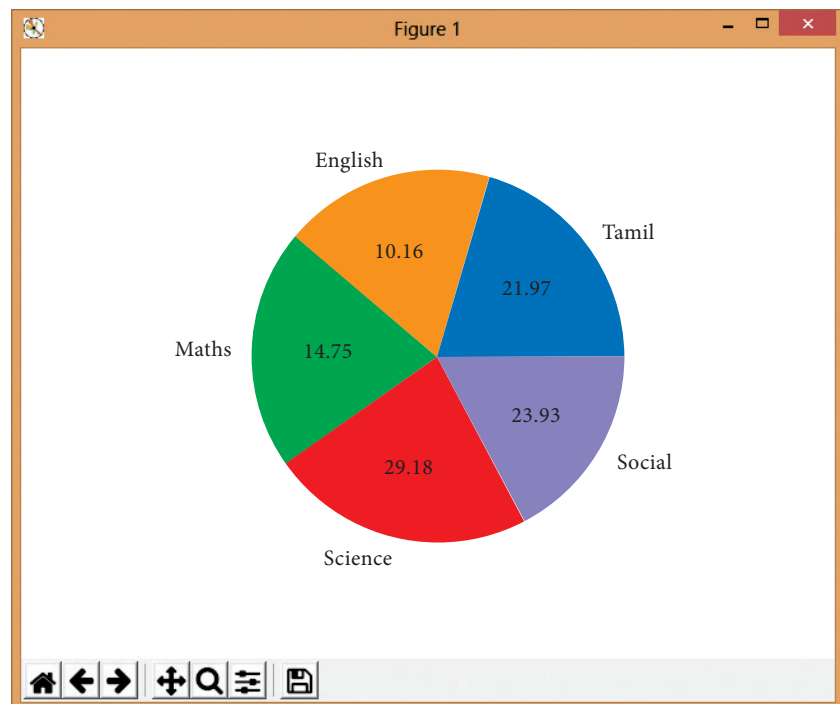
1.Tamil Mark = 67

2.English Mark = 31

3.Maths Mark = 45

4.Science Mark = 89

5.Social Mark = 73



INTERNAL CHOICES

Practical Question Number	Question 1		Question 2
CS1	PY1(a) Calculate Factorial PY1(b) Sum of Series	(OR)	PY9 - Python with SQL
CS2	PY2(a) Odd or Even PY2(b) Reverse the String	(OR)	PY8 - Python with CSV
CS3	PY3 - Generate values and remove odd numbers	(OR)	PY10 - Python Graphics with Pip
CS4	PY4 -Generate Prime numbers and Set Operations	(OR)	DB6 - MySQL – Employee Table
CS5	PY5 - Display a String elements – Using Class	(OR)	DB7 - MySQL – Student Table