COMPUTER SCIENCE PRACTICAL MANUAL

LIST OF PROGRAMS
CLASS – XII

All the programs start with the following code:

```
from datetime import datetime
import getpass
print("Date: ",datetime.now( ))
print("UserName: ",getpass.getuser( ))
```

Term-I

Class-XI Revision

P1. Take names and marks of 5 students and save as key: value pairs in a dictionary RESULT. WAP that prints the dictionary contents in ascending order of marks. (Do not use built in methods)

P2. WAP to print the following pattern:

```
********
      *
     *
   *
  *
********
```

P3. WAP to generate a **3 X 4 X 6** 3D array whose each element is *.

Functions

P4. Write a function to calculate volume of a box with appropriate default values for its parameters. Your function should have the following input parameters:

a) Length of box
b) Width of box
c) Height of box

Test it by writing complete program to invoke it.

P5. Write a program to have the following functions:

a) A function that takes a number as argument and calculates cube for it. The function does not return a value. If there is no value passed to the function in function call, the function should calculate cube of 2
b) A function that takes two char arguments and returns True if both the arguments are equal otherwise false.

Test both these functions by giving appropriate function call statements.

P6. Write a Python function to check whether a number is perfect or not (number to be tested as parameter).

In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself (also known as its aliquot sum). Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself).

Example: The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: (1 + 2 + 3 + 6) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128.

Test the function by giving appropriate function call statements.

P7. Write a Python program to execute a string containing Python code.

P8. Write a function that takes a number n and then returns a randomly generated number having exactly n digits(not starting with zero) e.g., if n is 2 then function can randomly return a number 10 – 99 but 07, 02 are not valid two digit numbers.

P9. Write a program that generates a series using a function which takes first and last values of the series and then generates four terms that are equidistant .g., if two numbers passed are 1 and 7 then function returns 1 3 5 7.

P10. Write a function that takes one argument (a positive integer) and reports if the argument is prime or not. Write a program that invokes this function.

Using Python Libraries

P11. Write a program with following function:

remove_letter(sentence, letter)

This function returns a copy of the above string with every instance of the indicated letter removed. E.g. , remove_letter("Welcome Everyone", "e") should return the string "Wlcom vryon".

P12. Create a module lengthconversion.py that stores functions for various length conversions:

miletokm( )
kmtomile( )
feettoinches( )
inchestofeet( )

Also store the constants – mileinkm = 1.609344, feetininches = 12.

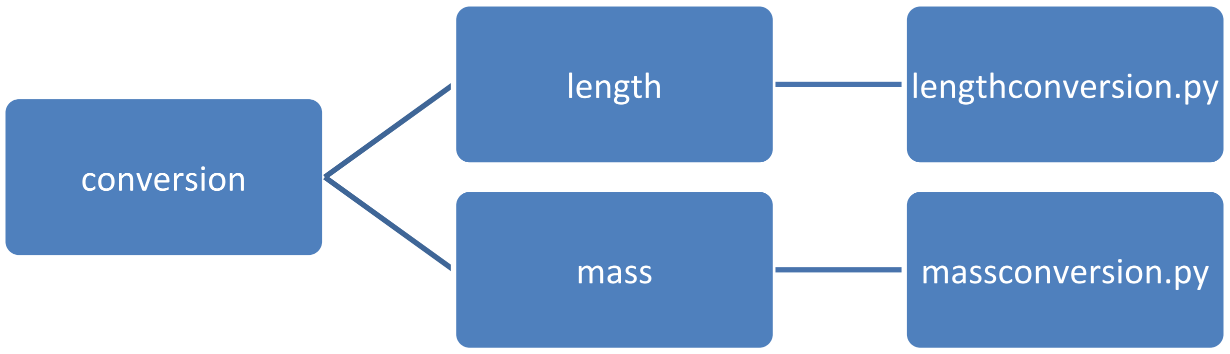
help( ) method should display proper information (to be shown as output)

P13. Create a module massconversion.py that stores functions for various mass conversions:

kgtotonne( )
tonnetokg( )
kgtopound( )
poundtokg( )

Also store the constants – kgintonne = 0.001, kginpound = 2.20462.
help( ) method should display proper information (to be shown as output)

P14. Create a package from the above two modules as this:



Make sure that above packages meet the requirements of being a Python Package.
Access at least two methods of lengthconversion.py using import statement.
Access at least two methods of massconversion.py using from-import statement.

Interface Python with MySQL

P15. Write a menu-driven program to do the following tasks:

1. Add a record
2. Add multiple records
3. Display all records
4. Update a record
5. Delete a record
6. Exit

Instructions for the above program:

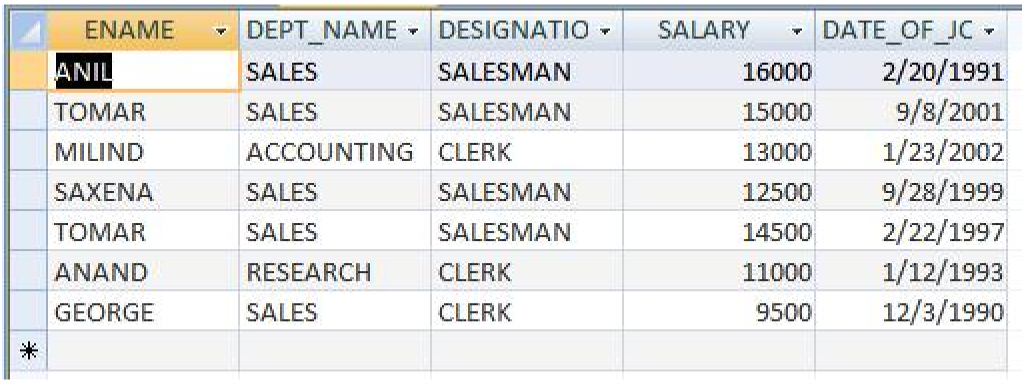
a) The program should ask the user for the choice number (1 to 6) every performed operation. The program should display the menu continually till the user enters 'no' as choice.

b) In every operation, the respective function should be called. So there should be 5 functions in total, each function should include creating a connection with database, performing the respective operation and closing the connection.

c) For all the operations, input should be taken from the user.

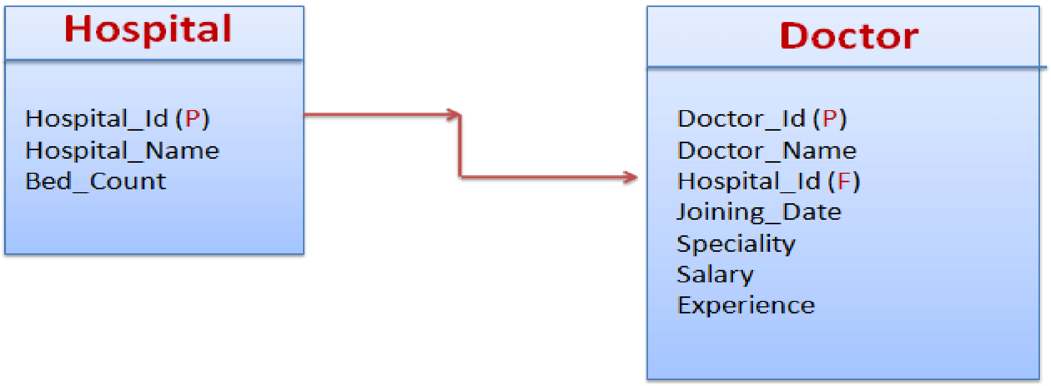
d) In update, insert and delete operations, you must use commit( ) and rollback( ).

e) Create the following table EMPLOYEE in your MySQL database: (Columns are: ENAME, DEPT_NAME, DESIGNATION, SALARY, DATE_OF_JOINING)

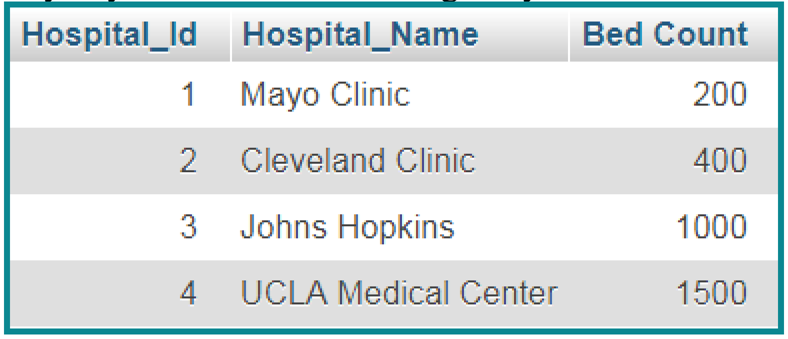


ENAME	DEPT_NAME	DESIGNATIO	SALARY	DATE_OF_JC
ANIL	SALES	SALESMAN	16000	2/20/1991
TOMAR	SALES	SALESMAN	15000	9/8/2001
MILIND	ACCOUNTING	CLERK	13000	1/23/2002
SAXENA	SALES	SALESMAN	12500	9/28/1999
TOMAR	SALES	SALESMAN	14500	2/22/1997
ANAND	RESEARCH	CLERK	11000	1/12/1993
GEORGE	SALES	CLERK	9500	12/3/1990

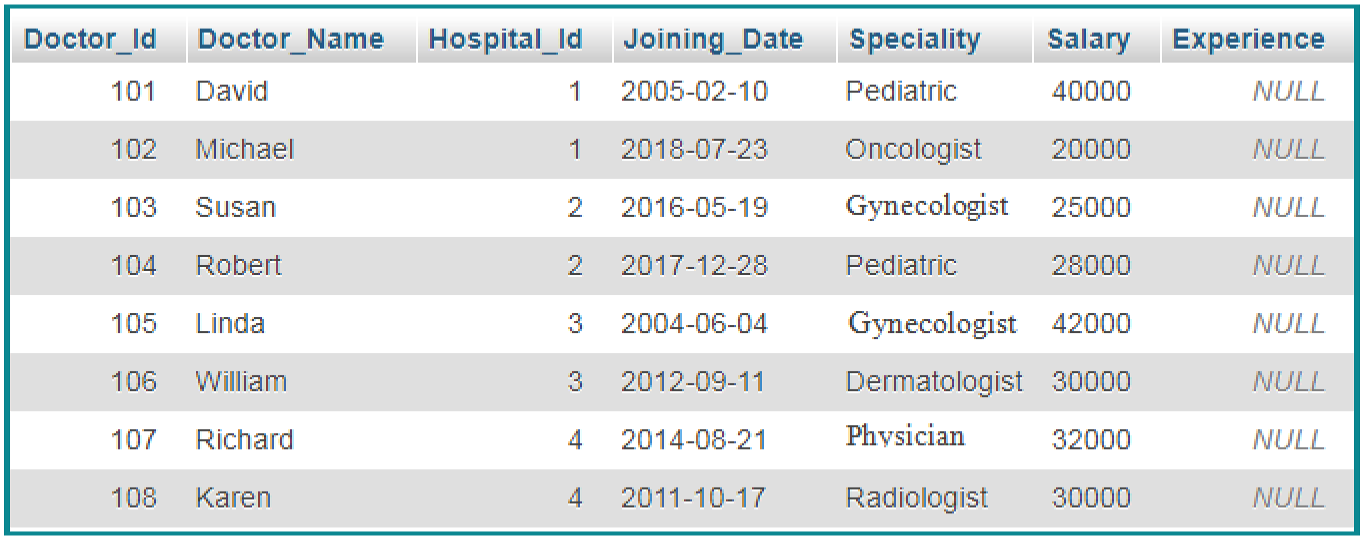
P16. Create the following data model in MySQL:



Where P indicates primary key and F indicates foreign key. The tables have following data:



Hospital_Id	Hospital_Name	Bed Count
1	Mayo Clinic	200
2	Cleveland Clinic	400
3	Johns Hopkins	1000
4	UCLA Medical Center	1500



Doctor_Id	Doctor_Name	Hospital_Id	Joining_Date	Speciality	Salary	Experience
101	David	1	2005-02-10	Pediatric	40000	NULL
102	Michael	1	2018-07-23	Oncologist	20000	NULL
103	Susan	2	2016-05-19	Gynecologist	25000	NULL
104	Robert	2	2017-12-28	Pediatric	28000	NULL
105	Linda	3	2004-06-04	Gynecologist	42000	NULL
106	William	3	2012-09-11	Dermatologist	30000	NULL
107	Richard	4	2014-08-21	Physician	32000	NULL
108	Karen	4	2011-10-17	Radiologist	30000	NULL

Write a program in Python to do the following:

a) Take Doctor_Id and Hospital_Id from the user and display all details from the respective tables

b) Display the list of doctors with their details for a given speciality (taken as input from user)

c) Display list of doctors with their details within a given hospital

** Part d) is not part of the file. You can always try to implement it. Use the hint and sample code to implement the same**

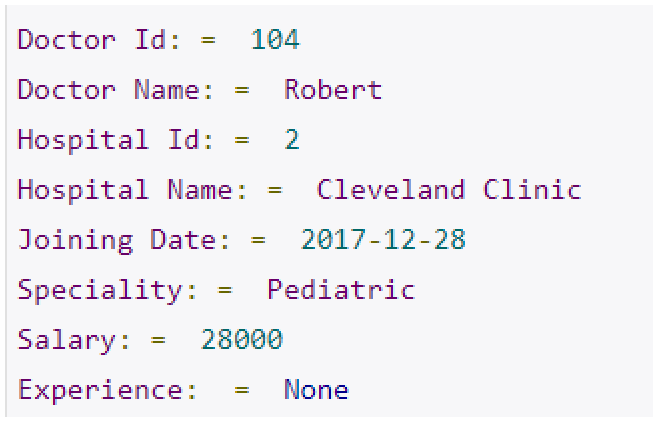
d) Implement the functionality to update the experience of each doctor in years (**Hint**: Doctor table has the joining date for each doctor. To get a difference in a year, we can calculate the difference between today's date and joining-date in years.) **Hint**:

```
# pip install py-dateutil

from datetime import datetime
from dateutil.relativedelta import relativedelta

d2 = datetime.now()
d1 = datetime.strptime(str("2019-4-14"),"%Y-%m-%d")
print(relativedelta(d2,d1).years) #d2, d1 should be datetime objects
```

The output for all parts should be in following format:
Sample Output:



```
Doctor Id: =  104
Doctor Name: =  Robert
Hospital Id: =  2
Hospital Name: =  Cleveland Clinic
Joining Date: =  2017-12-28
Speciality: =  Pediatric
Salary: =  28000
Experience:  =  None
```

Hint: To print the individual elements for printing, access them using indexing in the for loop.

CREATE FUNCTIONS AS AND WHEN REQUIRED

File Handling

17. WAP to read a text file line by line and display each word separated by a #.

18. Read a text file and display the number of vowels/ consonants/uppercase/ lowercase characters in the file.

19. WAP to count the words “to” and “the” present in a text file “Poem.txt”

20. Take a sample of ten phishing e-mails (or any text file) and find top two most commonly occurring words. (Way 1: Use d.items( ) as list and sort it to find top 2;

Way 2: Use max( ) method as follows:

```
>>> d = {'a':4, 'b':3, 'c':2, 'd':1}
>>> all_keys = list(d.keys())
>>> all_values = list(d.values())
>>> all_keys[all_values.index(max(all_values))])
```

21. WAP to remove all the lines that contain the article `a' in a file and write those lines (with ‘a’) to another file. (text files)

22. WAP to take the details of books from the user (title, price for each book) as many as he/she wants and write the records in text file. (e.g. of one record: Computer Science with Python, 500)

23. Write a function DISPLAYWORDS( ) in python to read lines from the text file STORY.TXT and display only those words, which are less than 4 characters.

24. WAP to read characters from the keyboard one by one till user enters “end” to finish the program. All lower case characters get stored in file LOWER.TXT, upper case in UPPER.TXT and all other characters in the file OTHERS.TXT.

25. WAP that reads a text file and creates another file that is identical except that every sequence of consecutive blank spaces is replaced by a single space.

26. WAP that reads a text file file1.txt and appends at the end of another file file2.txt, every line from file1.txt preceded by a line number.

27. WAP to find the size of file in bytes, number of words and number of lines

28. Create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message.

29. Create a binary file with roll number, name and marks. Input a roll number and update the marks.

30. Write a menu driven telephone directory program (Binary File handling) with the following options:

a) Add New Record
b) Display All Records
c) Search Telephone Number
d) Search Name
e) Update Telephone Number
f) Delete a record

{Hint: Take a dictionary PhoneBook with the following data members: name and phone_no. Add at least 5 records}

31. Create a binary file and write a string having two lines in it. Now read the file and display the text of string until the first letter 'o' is encountered. (Do not include 'o')

32. Considering the following definition of a dictionary MULTIPLEX, write a method in python to search and display all the content in a pickled file CINEMA.DAT, where MTYPE key of the dictionary is matching with the value 'Comedy'.

MULTIPLEX = {'MNO': ______, 'MNAME': __________, 'MTYPE':________}

33. WAP to increase the salary by Rs 2000/- of the employee having empno as 1251 in the file emp1.dat

34. WAP to get 5 items' details (itemno, name, price, category) from the user and create a CSV file (Items.csv)

Data for file:

IN_100	1K Ohm Resistor	88	Scientific
IN_101	Gopi Dairies	300	Books
IN_102	Study Table	4200	Furniture
IN_103	9V Battery	260	Batteries
IN_104	Bird House	450	Garden and Outdoors

35. WAP to read the file Items.csv and search for an item whose itemno is obtained from the user (as input)

36. WAP to read the file Items.csv and create a file highitems.csv, containing only those item details from Items.csv where price > 250.

37.WAP to create the same csv file as Q34 but with a delimiter character as pipe (|)

LISTS, STACKS and QUEUES:

38. Create a 2D list in Python that stores runs scored by a batsman in five overs:

Runs = [[0, 6, 4, 1, 0, 0], [3, 0, 2, 0, 0, 0], [0, 0, 4, 4, 0, 1], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0]]

a) WAF that returns the over in which the batsman scored the highest runs
b) WAF that returns the over(s) in which the batsman scored the minimum runs
c) WAF that returns the total runs scored by the batsman.

39. Write definition of a method AFIND(CITIES) to display all the city names from a list of cities, which are starting with alphabet ‘A’

40. Write a menu driven program to implement a stack for the students (adm_no, name). Perform the following operations on it:

a) Insert an element
b) Delete an element
c) Display the elements of stack

41. Write a function in Python, INSERTQ(Arr,data) and DELETEQ(Arr) for performing insertion and deletion operations in a Queue. Arr is the list used for implementing queue and data is the value to be inserted.

42. Write a function in python, MakePush(Package) and MakePop(Package) to add a new Package and delete a Package from a List of Package Description, considering them to act as push and pop operations of the Stack data structure.

43. WAP to print a string in reverse order. (Hint: Extract individual characters from string and push in stack; once done; keep popping from stack)

44. WAP to implement a queue of order numbers in a restaurant. Display a menu as shown below:

```
YUMMY PROGRAM

1. Order a Meal
2. Waiting Queue
3. Order is Ready
4. Exit
```

When option 1 is chosen, generate an order number of three digits, randomly and add it to your queue.

For option 2, display the orders that are in queue.

For option 3, dequeue an order number from the queue and flash it like:

Order Number 365 is ready

45. WAP to implement a stack of URLs in a web browser. Display a menu as shown below:

```
MANAGE HISTORY

1. Go to a new URL
2. URLs visited till now
3. Go back
4. Exit
```

When option 1 is chosen, ask the user for a URL and add it to stack

For option 2, display the URLs that are in stack.

For option 3, pop a URL and display the URL at top of the stack now with appropriate messages.