Practical File
Computer Science
(Python)

Name: Deepak Soni

Class : 12th (Science)

School: Kaliram Chandrakar Public School, Kurud

Roll No. : 12143

Index

CERTIFICATE

This is to certify that the contents of this project file submitted by **Deepak Soni** of class XII for the subject of Computer Science is his bonafide work submitted to **Kaliram Chandrakar Public School, Kurud** for partial fulfilment of the requirements for CBSE examinations of class XII.

The project has been completed under my guidance and supervision. **Deepak Soni** has been working on the completion of the project sincerely from start to finish. I certify that the project is up to my expectations and can be submitted for evaluation.

Sign. Of Principal

Sign. of Subject Teacher **Sign. External Examiner**

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my English teacher **Mr. Mohit Gedam Sir** as well as our principal **Mr. Devlal Yadav** who gave me the golden opportunity to do this wonderful project on the python programs and mysql.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Lastly, I like to thank all my supporters who have motivated me to fulfill their project before the timeline.

Date:08/01/2023

"Deepak Soni"

12th (science)

1.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
>>>
```

2. Program to enter two numbers and print the arithmetic operations like +,-,*, /, // and %.

Code:

```python
# Program for Arithmetic Calculator
result = 0
val1 = float(input("Enter the first value :"))
val2 = float(input("Enter the second value :"))
op = input("Enter any one of the operator (+,-,*,/,//,%)")
if op == "+":
    result = val1 + val2
elif op == "-":
    result = val1 - val2
elif op == "*":
    result = val1 * val2
elif op == "/":
    if val2 == 0:
        print("Please enter a value other than 0")
    else:
        result = val1 / val2
elif op == "//":
    result = val1 // val2
else:
    result = val1 % val2
print("The result is :", result)
```

Output:

```
========== RESTART: D:/New folder/Practi
Enter the first value :50
Enter the second value :24
Enter any one of the operator (+,-,*,/,//,%)+
The result is : 74.0
>>>
========== RESTART: D:/New folder/Practi
Enter the first value :50
Enter the second value :24
Enter any one of the operator (+,-,*,/,//,%)-
The result is : 26.0
>>>
========== RESTART: D:/New folder/Practi
Enter the first value :50
Enter the second value :24
Enter any one of the operator (+,-,*,/,//,%)*
The result is : 1200.0
>>>
========== RESTART: D:/New folder/Practi
Enter the first value :50
Enter the second value :24
Enter any one of the operator (+,-,*,/,//,%)/
The result is : 2.0833333333333335
>>>
========== RESTART: D:/New folder/Practi
Enter the first value :50
Enter the second value :24
Enter any one of the operator (+,-,*,/,//,%)//
The result is : 2.0
>>>
========== RESTART: D:/New folder/Practi
Enter the first value :50
Enter the second value :24
Enter any one of the operator (+,-,*,/,//,%)%
The result is : 2.0
>>>
```

3.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
>>>
================== RESTART:
Enter a number: 14
The given number is Even
>>>
```

4.Write a program to find whether an inputted number is perfect or not.

Code:

```
#   To find whether a number is perfect or not
def pernum(num):
    divsum=0
    for i in range(1,num):
        if num%i == 0:
            divsum+=i
    if divsum==num:
        print('Perfect Number')
    else:
        print('Not a perfect number')
pernum(6)
pernum(15)
```

Output:

```
Perfect Number
Not a perfect number
>>>
```

5.Write a Program to enter the number of terms and to print the Fibonacci Series.

```
# fibonacci
i = int(input("enter the limit:"))
x = 0
y = 1
z = 1
print("Fibonacci series \n")
print(x, y, end=" ")
while (z <= i):
    print(z, end=" ")
    x = y
    y = z
    z = x + y
```

Output:

```
enter the limit:50
Fibonacci series

0 1 1 2 3 5 8 13 21 34
>>>
```

6.Write a Program to enter the string and to check if it's palindrome or not using loop.

Code:

```
# Program to enter the string and check if it's palindrome or not using 'for' loop.
msg = input("Enter any string : ")
newlist = []
newlist[:0] = msg
l = len(newlist)
ed = l - 1
for i in range(0, l):
    if newlist[i] != newlist[ed]:
        print("Given String is not a palindrome")
        break
    if i >= ed:
        print("Given String is a palindrome")
        break
    l = l - 1
    ed = ed - 1
```

Output:

```
# Program to enter the string and check if it's palindrome or not using 'for' loop.
msg = input("Enter any string : ")
newlist = []
newlist[:0] = msg
l = len(newlist)
ed = l - 1
for i in range(0, l):
    if newlist[i] != newlist[ed]:
        print("Given String is not a palindrome")
        break
    if i >= ed:
        print("Given String is a palindrome")
        break
    l = l - 1
    ed = ed - 1
```

7. Write a Program to enter the number and print the Floyd's Triangle in decreasing order.

Code:

```
#Floyd's triangle
n=int(input("Enter the number :"))
for i in range(5,0,-1):
    for j in range(5,i-1,-1):
        print (j,end=' ')
    print('\n')
```

Output:

```
Enter the number :8
5

5 4

5 4 3

5 4 3 2

5 4 3 2 1

>>>
```

8. : Write a Program to find factorial of entered number using user-defined module fact().

Code:

```
#Module factfunc
def fact(no):
    while no > 0:
        f=f*no
        no=no-1
    return f
```

```
# Using function
import factfunc
x = int(input("Enter value for factorial : "))
ans = factfunc.fact(x)
print(ans)
```

Output:

```
Python 3.7.0 Shell
File  Edit  Shell  Debug  Options  Window  Help
Type "copyright", "credits" or "license()"
for more information.
>>>
 RESTART: C:\Users\preeti\AppData\Local\Pr
ograms\Python\Python37-32\prog_fact2.py
Enter value for factorial : 6
720
>>>
```

9. **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**

Code:

```
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:  {2, 3, 4, 5, 6}
Prime Numbers:  {2, 3, 4, 5, 7}
Union:  {2, 3, 4, 5, 6, 7}
Intersection:  {2, 3, 4, 5}
Difference:  {6}
Symmetric Difference:  {6, 7}
>>>
```

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

Code:

```
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
The given string contains...
2 Uppercase letters
15 Lowercase letters
7 Vowels
12 Consonants
0 Spaces
>>>
```

11.Write a Program to enter the numbers and find Linear Search, Binary Search, Lowest Number and Selection Sort using list/array code.

Code:

```
arr = []
def array_operation():
    ch = 1
    while ch != 10:
        print('Various Array operation\n')
        print('1 Create and Enter value\n')
        print('2 Print Array\n')
        print('3 Reverse Array\n')
        print('4 Linear Search\n')
        print('5 Binary Search\n')
        print('6 Lowest Number \n')
        print('7 Selection Sort\n')
        print('10 Exit\n')
        ch = int(input('Enter Choice '))
        if ch == 1:
            appendarray()
        elif ch == 2:
            print_array()
        elif ch == 3:
            reverse_array()
        elif ch == 4:
            linear_search()
        elif ch == 5:
            binary_search()
        elif ch == 6:
            min_number()
        elif ch == 7:
            selection_sort()
def appendarray():
    for i in range(0, 10):
        x = int(input('Enter Number : '))
        arr.insert(i, x)
```

```
# ------------------------------------------------------------------------------------------------------
def lsearch():
    try:
        x = int(input('Enter the Number You want to search : '))
        n = arr.index(x)
        print('Number Found at %d location'%(n + 1))
    except:
        print('Number Not Exist in list')
# ------------------------------------------------------------------------------------------------------
def linear_search():
    x = int(input('Enter the Number you want to search : '))
    fl = 0
    for i in range(0, 10):
        if arr[i] == x:
            fl = 1
            print('Number Found at %d location' % (i + 1))
            break
    if fl == 0:
        print('Number Not Found')
# ------------------------------------------------------------------------------------------------------
def binary_search():
    x = int(input('Enter the Number you want to search : '))
    fl = 0
    low = 0
    heigh = len(arr)
    while low <= heigh:
        mid = int((low + heigh)//2)
        if arr[mid] == x:
            fl = 1
            print('Number Found at %d location' % (mid + 1))
            break
        elif arr[mid] > x:
            low = mid + 1
        else:
            heigh = mid - 1
    if fl == 0:
        print('Number Not Found')
# ------------------------------------------------------------------------------------------------------
```

```
# ----------------------------------------------------------------------------------------------------

def min_number():
    n = arr[0]
    k = 0
    for i in range(0, 10):
        if arr[i] < n:
            n = arr[i]
            k = i
    print('The Lowest number is %d ' % (n))
# ----------------------------------------------------------------------------------------------------

def selection_sort():
    for i in range(0, 10):
        n = arr[i]
        k = i
        for j in range(i + 1, 10):
            if arr[j] < n:
                n = arr[j]
                k = j
                arr[k] = arr[i]
                arr[i] = n
array_operation()
```

Output:

```
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit
```

```
Enter Choice 1
Enter Number : 50
Enter Number : 20
Enter Number : 35
Enter Number : 86
Enter Number : 19
Enter Number : 9
Enter Number : 66
Enter Number : 96
Enter Number : 44
Enter Number : 56
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit
```

```
Enter Choice 2
50
20
35
86
19
9
66
96
44
56
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit
```

```
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit
```

```
Enter Choice 3
56
44
96
66
9
19
86
35
20
50
```

```
Enter Choice 4
Enter the Number you want to search : 19
Number Found at 5 location
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit
```

```
Enter Choice 5
Enter the Number you want to search : 19
Number Found at 5 location
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit
```

```
Enter Choice 6
The Lowest number is 9
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit
```

```
Enter Choice 7
Various Array operation

1 Create and Enter value

2 Print Array

3 Reverse Array

4 Linear Search

5 Binary Search

6 Lowest Number

7 Selection Sort

10 Exit

Enter Choice 10
>>>
```

12.Write a Program to read data from data file and show Data File Handling related functions utility in python

Code:

```
f = open("test.txt", 'r')
print(f.name)
f_contents = f.read()
print(f_contents)
f_contents = f.readlines()
print(f_contents)
f_contents = f.readline()
print(f_contents)
for line in f:
    print(line, end='')
f_contents = f.read(50)
print(f_contents)
size_to_read = 10
f_contents = f.read(size_to_read)
while len(f_contents) > 0:
    print(f_contents)
    print(f.tell())
    f_contents = f.read(size_to_read)
```

Output:

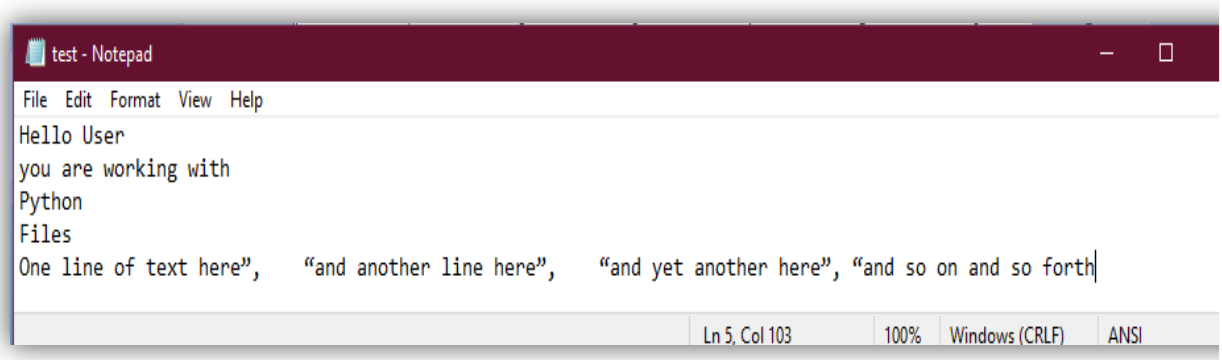
```
test.txt
Hello User
you are working with
Python
Files
[]


>>>
```

13.Write a Program to read data from data file in append mode and use writeLines function utility in python.

Code:

```
#Program to read data from data file in append mode
af=open("test.txt",'a')
lines_of_text = ("One line of text here",\
    "and another line here",\
    "and yet another here", "and so on and so forth")
af.writelines('\n' + lines_of_text)
af.close()
```

Output:



14. Write a Program to read data from data file in read mode and count the particular word occurrences in given string, number of times in python.

Code:

```
# Program to read data from data file in read mode and
# count the particular word occurrences in given string,
# number of times in python.
f = open("test.txt", 'r')
read = f.readlines()
f.close()
times = 0  # the variable has been created to show the number of times the loop ru
times2=0 #the variable has been created to show the number of times the loop runs
chk=input("Enter String to search : ")
count = 0
for sentence in read:
    line = sentence.split()
    times += 1
    for each in line:
        line2 = each
        times2 += 1
        if chk == line2:
            count += 1
print("The search String ", chk, "is present : ", count, "times")
print(times)
print(times2)
```

Output:

```
Enter String to search : Python
The search String  Python is present :  1 times
5
27
>>>
```

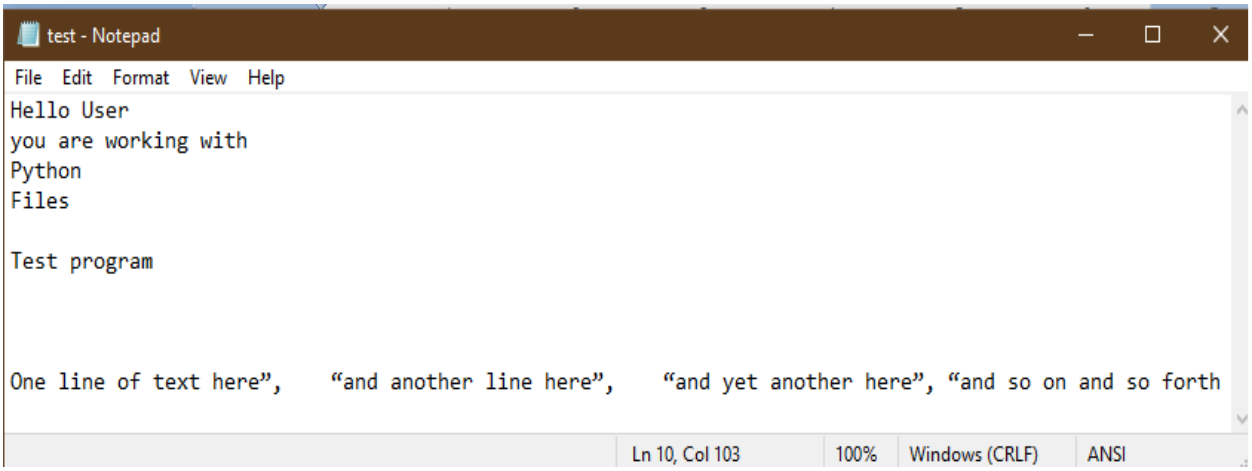
15.Write a Program to read data from data file in read mode and append the words starting with letter 'T' in a given file in python.

Code:

```
#Program to read data from data file in read mode and #append the words starting with letter 'T'
#in a given file in python
f=open("test.txt",'r')
read=f.readlines()
f.close()
id=[]
for ln in read:
    if ln.startswith("T"):
        id.append(ln)
print(id)
```

Output:

```
[]
>>>
========== RESTA
['Test program\n']
>>>
```



16. Write a Program to show MySQL database connectivity in python.

Code:

```
import mysql.connector
con=mysql.connector.connect(host='localhost',user='root',passwd='102318@#11',database='testdb')
stmt=con.cursor()
query='select * from emp;'
stmt.execute(query)
data=stmt.fetchone()
print(data)
```

Output:

```
(101, 'ANIL KUMAR', 'M', 'Programmer', Decimal('86000'))
>>>
```

17. Write a Python program to implement all basic operations of a stack, such as adding element (PUSH operation), removing element (POP operation) and displaying the stack elements (Traversal operation) using lists.

Code:

```
# Implementation of List as stack
s = []
c = "y"
while (c == "y"):
    print("1. PUSH")
    print("2. POP ")
    print("3. Display")
    choice = int(input("Enter your choice: "))
    if (choice == 1):
        a = input("Enter any number :")
        s.append(a)
    elif (choice == 2):
        if (s == []):
            print("Stack Empty")
        else:
            print("Deleted element is : ", s.pop())
    elif (choice == 3):
        l = len(s)
        for i in range(l - 1, -1, -1):  # To display elements from last element to first
            print (s[i])
    else:
        print("Wrong Input")
    c = input("Do you want to continue or not? ")
```

Output:

```
1. PUSH
2. POP
3. Display
Enter your choice: 1
Enter any number :9
Do you want to continue or not? y
1. PUSH
2. POP
3. Display
```

```
Enter your choice: 1
Enter any number :'d'
Do you want to continue or not? y
1. PUSH
2. POP
3. Display
Enter your choice: 1
Enter any number :5
Do you want to continue or not? y
1. PUSH
2. POP
3. Display
Enter your choice: 3
5
'd'
9
Do you want to continue or not? y
1. PUSH
2. POP
3. Display
Enter your choice: 2
Deleted element is :  5
Do you want to continue or not? n
>>>
```

18: Write a program to display unique vowels present in the given word using Stack.

Code:

```
#Program to display unique vowels present in the given word #using Stack
vowels =['a','e','i','o','u']
word = input("Enter the word to search for vowels :")
Stack = []
for letter in word:
    if letter in vowels:
        if letter not in Stack:
            Stack.append(letter)
print(Stack)
print("The number of different vowels present in",word,"is",len(Stack))
```

Output:

```
Enter the word to search for vowels :Hellopython
['e', 'o']
The number of different vowels present in Hellopython is 2
>>>
```

19. Write a program in Python to add, delete and display elements from a queue using list.

Code:

```
# Implementing List as a Queue - using function append() and pop()
a = []
c = 'y'
while (c == 'y'):
    print("1. INSERT")
    print("2. DELETE ")
    print("3. Display")
    choice = int(input("Enter your choice: "))
    if (choice == 1):
        b = int(input("Enter new number: "))
        a.append(b)
    elif (choice == 2):
        if (a == []):
            print("Queue Empty")
        else:
            print("Deleted element is:", a[0])
            a.pop(0)

    elif (choice == 3):
        l = len(a)
        for i in range(0, l):
            print(a[i])
    else:
        print("wrong input")
    c = input("Do you want to continue or not: ")
```

Output:

```
1. INSERT
2. DELETE
3. Display
Enter your choice: 1
Enter new number: 45
Do you want to continue or not: y
1. INSERT
2. DELETE
3. Display
Enter your choice: 1
Enter new number: 6
Do you want to continue or not: y
1. INSERT
2. DELETE
3. Display
Enter your choice: 1
Enter new number: 16
Do you want to continue or not: y
1. INSERT
2. DELETE
3. Display
Enter your choice: 3
45
6
16
Do you want to continue or not: y
1. INSERT
2. DELETE
3. Display
Enter your choice: 2
Deleted element is: 45
Do you want to continue or not: n
>>>
```

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

Code:

```
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:Andrew Green
Enter Age:23
Enter Name:Karan Mahanand
Enter Age:17
Enter Name:Divya Sharma
Enter Age:18
Enter Name:Jane Foster
Enter Age:30
Enter Name:Neha Agrawal
Enter Age:16
Enter Name:Sahil Kashyap
Enter Age:32
Enter Name:Rohan Mishra
Enter Age:20
Enter Name:Leena Sahu
Enter Age:22
Enter Name:Ajay Kumar
Enter Age:19
Enter Name:Swati Singh
Enter Age:22
```

```
Displaying All the Records From student Table in Descending order of age
('Sahil Kashyap', 32)
('Jane Foster', 30)
('Andrew Green', 23)
('Leena Sahu', 22)
('Swati Singh', 22)
('Rohan Mishra', 20)
('Ajay Kumar', 19)
('Divya Sharma', 18)
('Karan Mahanand', 17)
('Neha Agrawal', 16)
>>>
```

21. 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 Outputs:

(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         | NO   | PRI | NULL    |       |
| Empname | varchar(20) | YES  |     | NULL    |       |
| Desig   | varchar(10) | YES  |     | NULL    |       |
| Dept    | varchar(10) | YES  |     | NULL    |       |
| Age     | int         | YES  |     | NULL    |       |
| Place   | varchar(10) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
6 rows in set (0.01 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 | Siddharth | Officer | Accounts |   45 | Salem      |
|  1222 | Naveen    | Manager | Admin    |   32 | Erode      |
|  1223 | Ramesh    | Clerk   | Accounts |   33 | Ambathur   |
|  1224 | Abhinaya  | 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');
Query OK, 1 row affected (0.00 sec)

mysql> Insert into employee values(3227, 'Rekha', 'Officer','Admin',34,'Salem');
Query OK, 1 row affected (0.00 sec)

mysql> Select * from Employee;
+-------+-----------+---------+----------+------+------------+
| Empno | Empname   | Desig   | Dept     | Age  | Place      |
+-------+-----------+---------+----------+------+------------+
|  1221 | Siddharth | Officer | Accounts |   45 | Salem      |
|  1222 | Naveen    | Manager | Admin    |   32 | Erode      |
|  1223 | Ramesh    | Clerk   | Accounts |   33 | Ambathur   |
|  1224 | Abhinaya  | 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);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> Desc Employee;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Empno   | int         | NO   | PRI | NULL    |       |
| Empname | varchar(20) | YES  |     | NULL    |       |
| Desig   | varchar(10) | YES  |     | NULL    |       |
| Dept    | varchar(10) | YES  |     | NULL    |       |
| Age     | int         | YES  |     | NULL    |       |
| Place   | varchar(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 | Siddharth | 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 | Abhinaya | 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)
```

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

```
mysql> select * from employee where doj>'2018-01-01';
+-------+----------+---------+----------+------+-----------+------------+
| Empno | Empname  | Desig   | Dept     | Age  | Place     | doj        |
+-------+----------+---------+----------+------+-----------+------------+
|  1224 | Abhinaya | 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)
```

22. 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) Modify the class M2 to M1.

Check for the uniqueness of Register no.

SQL Queries and Outputs:

(1) Creating Table – Student:

```
mysql> Create table Student(
    -> Reg_No char(5),
    -> Sname varchar(20),
    -> Age integer(2),
    -> Dept varchar(10),
    -> Class char(3));
Query OK, 0 rows affected, 1 warning (0.04 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         | YES  |     | NULL    |       |
| Dept   | varchar(10) | YES  |     | NULL    |       |
| Class  | char(3)     | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
5 rows in set (0.01 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 (0.00 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 (0.00 sec)
```

(iii) Modify the class M2 to M1:

```
mysql> Update Student set Class='ME1' where Class='ME2';
Query OK, 1 row affected (0.00 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   | ME1   |
| 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)
```

(iv) Check for the uniqueness of Register no:

```
mysql> Select Distinct Reg_No from Student;
+--------+
| Reg_No |
+--------+
| M1001  |
| M1002  |
| C1001  |
| C1002  |
| E1001  |
| E1002  |
| E1003  |
+--------+
```

23. write a Python Program to integrate MYSQL with Python by inserting records to Emp table and display the records.

Source code:

```
import mysql.connector
con=mysql.connector.connect(host='localhost',username='root',password='root',database='employees')
if con.is_connected():
   cur=con.cursor()
   opt='y'
   while opt=='y':
         No=int(input("Enter Employee Number:"))
         Name=input("Enter Employee Name:")
         Gender=input("Enter Employee Gender(M/F):")
         Salary=int(input("Enter Employee Salary:"))
         Query="INSERT INTO EMP VALUES({},'{}','{}',{})".format(No,Name,Gender,Salary)
         cur.execute(Query)
         con.commit()
         print("Record Stored Successfully")
         opt=input("Do you want to add another employee details(y/n):")

Query="SELECT * FROM EMP";
cur.execute(Query)
data=cur.fetchall()
for i in data:
    print(i)
con.close()
```

SAMPLE OUTPUT:

Python Executed Program Output:

```
Enter Employee Number:1
Enter Employee Name:Arun
Enter Employee Gender(M/F):M
Enter Employee Salary:20000
Record Stored Successfully
Do you want to add another employee details(y/n):y
Enter Employee Number:2
Enter Employee Name:Bala
Enter Employee Gender(M/F):M
Enter Employee Salary:25000
Record Stored Successfully
```

```
Do you want to add another employee details(y/n):y
Enter Employee Number:3
Enter Employee Name:Bavya
Enter Employee Gender(M/F):F
Enter Employee Salary:27000
Record Stored Successfully
Do you want to add another employee details(y/n):y
Enter Employee Number:4
Enter Employee Name:Saravanan
Enter Employee Gender(M/F):M
Enter Employee Salary:29000
Record Stored Successfully
Do you want to add another employee details(y/n):n
(1, 'Arun', 'M', 20000)
(2, 'Bala', 'M', 25000)
(3, 'Bavya', 'F', 27000)
(4, 'Saravanan', 'M', 29000)
```

SQL OUTPUT:

```
mysql> SELECT * FROM EMP;
+-------+-----------+--------+--------+
| EMPID | NAME      | GENDER | SALARY |
+-------+-----------+--------+--------+
|     1 | Arun      | M      |  20000 |
|     2 | Bala      | M      |  25000 |
|     3 | Bavya     | F      |  27000 |
|     4 | Saravanan | M      |  29000 |
+-------+-----------+--------+--------+
4 rows in set (0.00 sec)
```

24. CREATE A PYTHON PROGRAM TO INTEGRATE MYSQL WITH PYTHON (SEARCH AND DISPLAY RECORDS)

Source code:

```
import mysql.connector
con=mysql.connector.connect(host='localhost',username='root',password='root',database='employees')
if con.is_connected():
   cur=con.cursor()
```

```
    print("*********************************")
    print("Welcome to Employee Search Screen")
    print("*********************************")
    No=int(input("Enter the employee number to search:"))
    Query="SELECT * FROM EMP WHERE EMPID={}".format(No)
    cur.execute(Query)
    data=cur.fetchone()
    if data!=None:
            print(data)
    else:
            print("Record not Found!!!")
con.close()
```

SAMPLE OUTPUT:

Python Executed Program Output:

RUN -1:

```
*********************************
Welcome to Employee Search Screen
*********************************
Enter the employee number to search:2
(2, 'Bala', 'M', 25000)
```

Run -2:

```
*********************************
Welcome to Employee Search Screen
*********************************
Enter the employee number to search:505
Record not Found!!!
```

SQL OUTPUT:
OUTPUT -1:

```
mysql> SELECT * FROM EMP WHERE EMPID=2;
+-------+------+--------+--------+
| EMPID | NAME | GENDER | SALARY |
+-------+------+--------+--------+
|     2 | Bala | M      |  25000 |
+-------+------+--------+--------+
1 row in set (0.00 sec)
```

OUTPUT -2:

```
mysql> SELECT * FROM EMP WHERE EMPID=505;
Empty set (0.00 sec)
```

25. write a Python Program to integrate MYSQL with Python to search an Employee using EMPID and update the Salary of an employee if present in already existing table EMP, if not display the appropriate message.

Source code:

```
import mysql.connector
con=mysql.connector.connect(host='localhost',username='root',password='root',database='employees')
if con.is_connected():
    cur=con.cursor()
    print("****************************************")
    print("Welcome to Employee detail update Screen")
    print("****************************************")
    No=int(input("Enter the employee number to Update:"))
    Query="SELECT * FROM EMP WHERE EMPID={}".format(No)
    cur.execute(Query)
    data=cur.fetchone()
    if data!=None:
            print("Record found details are:")
            print(data)
            ans=input("Do you want to update the Salary of the above employee(y/n)?:")
            if ans=='y' or ans=='Y':
                New_Sal=int(input("Enter the New Salary of an Employee:"))
                Q1="UPDATE EMP SET SALARY={} WHERE EMPID={}".format(New_Sal,No)
                cur.execute(Q1)
                con.commit()
                print("Employee Salary Updated Successfully")
                Q2="SELECT * FROM EMP"
                cur.execute(Q2)
                data=cur.fetchall()
                for i in data:
                    print(i)
    else:
       print("Record not Found!!!")
```

SAMPLE OUTPUT:

Python Executed Program Output:

RUN -1:

```
****************************************
Welcome to Employee detail update Screen
****************************************
Enter the employee number to search:3
Record found details are:
(3, 'BAVYA', 'F', 27000)
Do you want to update the Salary of the above employee(y/n)?:y
Enter the New Salary of an Employee:30000
Employee Salary Updated Successfully
(1, 'Arun', 'M', 20000)
(2, 'Bala', 'M', 25000)
(3, 'BAVYA', 'F', 30000)
(4, 'Saravanan', 'M', 29000)
```

Run -2:

```
****************************************
Welcome to Employee detail update Screen
****************************************
Enter the employee number to Update:500
Record not Found!!!
```

SQL OUTPUT:

```
mysql> SELECT * FROM EMP WHERE EMPID=3;
+-------+-------+--------+--------+
| EMPID | NAME  | GENDER | SALARY |
+-------+-------+--------+--------+
|     3 | BAVYA | F      |  27000 |
+-------+-------+--------+--------+
1 row in set (0.00 sec)
```

```
mysql> UPDATE EMP SET SALARY=30000 WHERE EMPID=3;
Query OK, 1 row affected (0.29 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
+-------+-----------+--------+--------+
| EMPID | NAME      | GENDER | SALARY |
+-------+-----------+--------+--------+
|     1 | Arun      | M      |  20000 |
|     2 | Bala      | M      |  25000 |
|     3 | BAVYA     | F      |  30000 |
|     4 | Saravanan | M      |  29000 |
+-------+-----------+--------+--------+
```