

CBCS SCHEME

USN

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15CS664

Sixth Semester B.E. Degree Examination, Aug./Sept. 2020
Python Application Programming

Time: 3 hrs.

Max. Marks: 80

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

- 1 a. Explain types of error with examples. (04 Marks)
 b. Explain various Names, Keywords and expressions with examples. (06 Marks)
 c. Write a python program using try and except, so that your program handles non-numeric input gracefully by printing a message and exiting the program the following shown two execution of the program
 Enter Hours : 20
 Enter Rate : nine
 Error, please enter numeric input
 Enter hours : forty
 Error, please enter numeric input. (06 Marks)

OR

- 2 a. Explain conditional execution, Alternative execution chained conditionals and nested conditionals with examples. (08 Marks)
 b. Explain break and continue statement with examples in python. (04 Marks)
 c. Explain with an example what are fruitful functions and void functions. (04 Marks)

Module-2

- 3 a. Explain while and for loops with examples. (04 Marks)
 b. Write a python program to find the largest value from the given set of accepted values. (06 Marks)
 c. Discuss the string handling methods in python with examples. (06 Marks)

OR

- 4 a. Write a python program to check whether a given string is palindrome or not. (06 Marks)
 b. Explain with example the syntax of read (), write () methods for a file. (04 Marks)
 c. Develop a python program for creating a copy an existing file. (06 Marks)

Module-3

- 5 a. Explain the difference between a list and a dictionary. (04 Marks)
 b. Make a list of first ten letters of the alphabet then using the slice operation do the following:
 i) Print the first three letters from the list
 ii) Print any three letters from the middle
 iii) Print the letters from 5th letters to the end of the list. (04 Marks)
 c. Discuss the lists handling functions in python with example. (08 Marks)

1 of 2

ALL BRANCHES | ALL SEMESTERS | NOTES | QUESTION PAPERS | LAB MANUALS

A Vtresource Go Green initiative

Head of the Department
 Dept. of Electronic & Communication Engg.
 KJ Somaiya Institute of Technology
 22.07.2021

Prepared & Compiled By
 Prof. S.S.

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
 2. Any revealing of identification, appeal to evaluator and /or equations written eq. 42+8 = 50, will be treated as malpractice

15CS664

OR

- 6 a. Differentiate between list and dictionary. (08 Marks)
b. Define tuple, explain DSU pattern. Write a python code to determinate tuples by sorting a list of words from longest to shortest using loops. (04 Marks)
c. Explain the need of Regular expressions in python language. (04 Marks)

Module-4

- 7 a. Explain classes and attributes in python language with examples. (05 Marks)
b. Explain pure functions and modifiers with examples. (05 Marks)
c. Write a program that uses class to store the name and marks of students. Use list to store the marks in three subjects. (06 Marks)

OR

- 8 a. Explain initialization method with example. (04 Marks)
b. Write a class Rectangle that has attributes length and breadth and a method area which returns the area of the rectangle. (06 Marks)
c. What is operator overloading? Write python code to overload "+" "-" and "*" operator by providing the methods __add__, __sub__ and __mul__. (06 Marks)

Module-5

- 9 a. Write a python code for retrieving the romeo.txt file from the web and compute the frequency of each word in the file. (06 Marks)
b. Write a note on XML. (05 Marks)
c. Explain with a neat diagram of Service Oriented Architecture. (05 Marks)

OR

- 10 a. Describe creation of database table using database cursor architecture. (08 Marks)
b. Write a python code for creating employee database, inserting records and selecting the employees working in the company. (08 Marks)

Name of the faculty: Prof. Shree Gowri SS

Name of Institution: KLS VJIT Haliyal

Department: Electronics and Communication

Sem : Sixth semester B.E Degree Examination
Aug/sept 2020

Subject : Python Application Programming

Subject Code : 15CS664

Prof. 20/07/2021

Subject Teacher

(Prof. Shree Gowri.S.S)

Module - 1

1a. Explain types of errors with examples — 04 Marks

Answer: There are three types of error & they are

1) Syntax error:- The statements that are not following the grammar of the programming language

2) Logical error:- A logical error occurs when syntax is correct but there is mistake in the order of the statement i.e. order of statement is not proper.

3) Semantic error:- A semantic error is when the statement is syntactically correct and order is also perfect but there is a simple mistake in program i.e. the program is perfectly correct but it does not do what you intended for it to do.

1b. Explain various Names, keywords and expressions with examples — 06 marks

Answer Names:- A name in python is called as identifier.

It can be a variable name, class name, function name, object name or module name.

Naming (identifiers) follow some rules

1) Should not start with digit

2) identifiers are case sensitive

3) keywords/reserve words cannot be used

4) Special symbols are not used like @, #, \$, %

Examples for Names

Page - 2

Ami
(Prof. Shree Gans)

- 1) abc
- 2) total123
- 3) abc-def

Keywords :- The interpreter uses keywords to recognize the structure of the program. Python reserves 33 keywords.

few of Python keywords are

and def elif is try break return for
as if else in block continue while with

Expression : Expression is a combination of values, variables and operators.

Syntax :- variable_name = expression

i.e. assigning value to the variable

The operand to the left of equal sign is called as variable & the operand to the right of equal sign is called as expression.

Ex :- $x = 5$

- c. write a python program using try and except so that your program handle non-numeric input gracefully by printing a message and exiting the program the following shows two execution of the program

— 06 marks

Enter Hours: 20

Enter Rate: nine

Error, please enter numeric input

Enter Hours: forty

Error, please enter numeric input

Answer: let us compute the gross pay by using hours and rate parameter for a working employee i.e. $\text{pay} = \text{hours} \times \text{rate}$

In order to compute this the user need to enter the numeric value for hours and rate. Suppose if non-numeric data is entered program end gracefully by printing an appropriate message like "Please enter numeric input".

Program:

```
hours = float(input("Enter the hours: "))
```

```
rate = float(input("Enter the rate: "))
```

```
try :
```

```
    pay = hours * rate
```

```
    print(pay)
```

```
except :
```

```
    print("Please enter numeric input")
```

o/p:

Enter hours : 20

Enter rate : nine

Error, Please enter numeric input

Enter hours : forty

Error, Please enter numeric input.

Q.1. Explain conditional execution, alternative chained conditionals and nested conditionals with examples — 08 Marks

Answer: conditional execution code

i.e. if decision control flow statement

~~i.e. if... else decision control flow statement~~

~~i.e. if... elif... else decision code~~

The simplest form is if statement

```
if x > 0:
```

```
    print('x is positive')
```

i.e. Syntax for if statement is

```
if boolean_expression:
```

```
    statement
```

Alternative execution

Second form of the if statement is alternative execution, in which there are two possibilities

The syntax is

```
if boolean_expression:
```

```
    statement_1
```

```
else:
```

```
    statement_2
```

Example:

```
if x > 0:
```

```
    print('x is positive')
```

```
else:
```

```
    print('x is negative')
```

Chained conditionals

Page-5

Dr.
(Prof. Shree
Gowari)

Chained conditionals are used when there are more than two possibilities

Syntax:

```
if boolean-exp1:  
    Statement_1  
  
elif boolean-exp2:  
    Statement_2  
  
elif boolean-exp3:  
    Statement_3  
  
⋮  
⋮  
else:  
    last_statement
```

Example:

```
if x < y:  
    print ("x is less than y")  
  
elif x > y:  
    print ('x is greater than y')  
  
else:  
    print ('x and y are equal')
```

Nested conditionals: one condition can also be nested inside another i.e. an if statement contains another if statement either in if blocks or in else blocks

Syntax of Nested conditional Statement

```

if boolean_exp1:
    if boolean_exp_a:
        Statement_1
    else:
        Statement_2
else:
    Statement_last

```

If the boolean_exp1 is evaluated to True the control is transferred to boolean_exp_a and it is evaluated. If it evaluates to True Statement_1 is executed else Statement_2 is executed. If boolean_exp1 is evaluated to False then Statement_last is executed.

2b Explain break and continue statement with examples in python — 04 Marks.

Answer: break Statement: break is a keyword in python. break Statement is useful in case of loops i.e. when loop is going to become an infinite loop. In order to come out from the infinite loop break Statement is used.

Syntax

```

while testexpression:
    if condition:
        break

```

Continue Statement : breaks statement helps to come out from the infinite loop. where as continue statement is used to skip the current iteration of the loop.

Qc. Explain with an example what are the fruitful function and void functions - 04 marks

Answer: fruitful function :- some of the functions return results are called fruitful functions

Example : $x = \text{math.cos}(\text{radians})$
 $\text{golden} = (\text{math.sqrt}(5) + 1) / 2$

when fruitful function is called in interactive mode then python displays the result.

But in a script mode, fruitful function do not store the result of the function in a variable & hence the returned value is vanished.

Void function :- void function does not return a value. If the function do not return any value we call that function as void function

Example :

```
def print_twice(spam):  
    print(spam)  
    print(spam)  
  
x = print_twice(17)
```

O/P: 17
17
print(x)
None

Module - 2

3a. Explain while and for loops with examples

04 - Marks

Answer: while loop

Syntax:

while boolean-expression:

Statement_1

Statement_2

⋮

Statement - n

Statement - after - while

while loop starts with

1) while keyword

2) boolean-expression is specified & ends with colon

3) consist block of statement with indentation

The flow of execution for while statement is as follows

1. Evaluates the boolean expression, the evaluation of condition results either True or False
2. If the condition is False, exit the while statement i.e. block of statement is not executed
3. If the condition is True, it executes block of statement (i.e. body of loop) and again go back to step 1 & continues

for loop

Syntax of for loop :

```
for iteration_variable in sequence:  
    Statement_1  
    Statement_2  
    ;  
    Statement_n  
Statement_after_for_loop
```

for loop starts with

1. for loop keyword and ends with colon
2. iteration_variable is the iteration_variable
3. The iteration_variable can be any valid variable name
4. Sequence is a set of elements or items on which loop is iterated.

The process of assigning items from the sequence to the iteration_variable & the executing the body of the loop continues until all the elements or items in the sequence are completed.

Example

```
names = ["ECE", "VTU", "Belgum"]
```

```
for a in names:
```

```
    print("welcome", a)
```

```
print("Done")
```

o/p: Welcome FLE
Welcome VTU
welcome Belgium
~~@one~~

Example for while loop

```
while True:
    x = int(input("Enter a number: "))
    if x > 0:
        print("The number is positive")
    else:
        print("The number is negative")
        break
```

36 write a python program to find the largest value from the given set of accepted values
- 06 Marks

Program

```
largest = None
print("Before finding largest number largest is: ", largest)
for i in [3, 41, 21, 9, 74, 12, 15]:
    if largest is None or i > largest:
        largest = i
    print("Iteration is: ", i, " and largest is: ", largest)
print("largest number in list: ", largest)
```


Iteration is: 37 and largest is: 3

Iteration is: 41 and largest is: 41

Iteration is: 21 and largest is: 41

Iteration is: 9 and largest is: 41

Iteration is: 74 and largest is: 74

Iteration is: 12 and largest is: 74

Iteration is: 15 and largest is: 74

3c. Discuss the string handling methods in Python with examples ← 06 marks.

Answer: String Methods

1) upper(): The method upper takes a string and returns a new string with all uppercase letters.

Ex:-
 >>> word = "oops"
 >>> newword = word.upper()
 >>> print(newword)

OOPS

2) lower():- The method lower() converts upper case to lower case

Example :- >>> "EXAM".lower()
 exam

3) capitalize():- This method returns a string with its first character capitalized and and remaining are lowercase

example :- Quote = " hi how Ru You "

Quote.capitalize()

Hi how ru you.

Page -
51
(Prof. Sir
Gow)

4) find(): - checks if substring appears in the string name and returns position of first character of substring. returns -1 if substring not found

```
>>> "Cucumber".find("cu")
```

0

```
>>> "Cucumber".find("xyz")
```

-1

5) count(): - This method returns the number of occurrence of substring in string

```
>>> language = "Monty Python"
```

```
>>> language.count("n")
```

2

OR

4a Write a python program to check whether a given string is palindrome or not — 06 mark

Program

```
def palindrome (user_string):
```

```
    if user_string == user_string[::-1]:
```

```
        print("user entered string is palindrome")
```

```
    else:
```

```
        print("user entered string is not  
palindrome")
```

```
input_string = input("Enter a string: ")
palindrome(input_string)
```

Output

Enter a string: madam

user entered string is palindrome

Enter a string: college

user entered string is not palindrome.

4b Explain with example the syntax of read(), write() methods for a file — 04 marks

Answer: read() method returns the file content

Syntax:

objectname.method()

Example:

```
f = open("filename.txt")
```

```
print(f.read())
```

write() method :- writes the specified string to the file

Syntax:-

objectname.write()

Example:-

```
fhand = open("filename.txt")
```

```
line1 = "hi how are you"
```

```
fhand.write(line1)
```

```
fhand.close()
```


4c Develop a python program for creating a copy of an existing file. — 06 Marks

Program

```
>>> import shutil
>>> shutil.copy('original.txt', 'duplicate.txt')
>>> shutil.copy('original.txt', 'my_folder/duplicate.txt')
```

Module - 3

5a. Explain the difference between list and a dictionary — 04 marks

Answer:- The difference between list and dictionary

List

dictionary

- 1) list is an ordered sequence
- 2) The items of list are accessed using index position

- 1) dictionaries are unordered sets
- 2) They are accessed via key

3) list is created by placing elements in [] separated by comma

3) Dictionary is created by placing elements ↓ of ↓ as
in a "key": "value"

5b. Make a list of first ten letters of the alphabet then using the slice operation do the following;

i) print the first three letters from the list

ii) print any three letters from the middle

iii) print any three the letters from 5th letters to the end of the list

— 04 marks

answer:- Program

```
>>> a = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

```
>>> x = a[0:3]
```

```
>>> print(x)
```

```
['a', 'b', 'c']
```

```
>>> x = a[3:6]
```

```
>>> print(x)
```

```
['d', 'e', 'f']
```

```
>>> x = a[5:]
```

```
['f', 'g', 'h', 'i', 'j']
```

5C. Discuss the list handling functions in python with example — 08 marks

Answer:

1) append() → adds new element to the end of the list

```
>>> t = ['a', 'b', 'c', 'd']
```

```
>>> t.append('e')
```

```
>>> print(t)
```

```
['a', 'b', 'c', 'd', 'e']
```

2) clear() → clears the list

```
>>> t = ['a', 'b', 'c', 'd', 'e']
```

```
>>> t.clear()
```

```
>>> print(t)
```

```
[]
```

3) extend() :- This method takes list as an argument and adds this list at the end of the current list

```
>>> t1 = ['a', 'b', 'c']
```

```
>>> t2 = ['d', 'e']
```

```
>>> t1.extend(t2)
```

```
>>> print(t1)
```

```
['a', 'b', 'c', 'd', 'e']
```

```
>>> print(t2)
```

```
['d', 'e']
```


4) POP():- pop() method is used to delete the element from the list. This method is used if we know that the index of the element that we need to delete.

pop() method removes the element from the list at the specified position. This method deletes and returns the element that was deleted or removed.

```
>>> t = ['a', 'b', 'c']
```

```
>>> x = t.pop(1)
```

```
>>> print(t)
```

```
>>> print(x)
```

```
['a', 'c']
```

```
b
```

OR

6a. Differentiate between list and dictionary - 8 marks

Answer:

List

1. list is a sequence of values/elements

2. ex:- l = []

3. Creating a list
l = list()

4. lists are mutable

5. Accessing elements of list using integer index

Dictionary

1. Dictionary ^{consists} of key-value pair

2. d = { }

3. creating a dictionary
d = dict()

4. dictionary are mutable

5. Accessing items of dictionary using key

66. Define Tuple, explain DSU pattern. Write a Python code to determine tuples by sorting list of words from longest to shortest using loops

04 marks

Answer: A tuple is a sequence of values like a list. The values stored in a tuple can be of any type and they are indexed by integers. Tuple is a comma separated list of values.

DSU :- Decorate sort Undecorate

DSU is a pattern that involves building a list of tuples, sort and extracting part of the result.

Decorate: building list of tuples

Sort: sort the list of tuples

Undecorate: extracting the part of sorted elements of the sequence.

Program

```
txt = ' I am more beautiful'
```

```
words = txt.split()
```

```
t = list()
```

```
for word in words:
```

```
    t.append((len(word), word))
```

```
t.sort(reverse=True)
```

```
res = list()
```

```
for length, word in t:
```

```
    res.append(word)
```


print(res)

['beautiful', 'more', 'am', 'I']

3c Explain the need of ↓ Python language — 04 marks
Regular Expression in

Answer: Regular Expression are essentially a tiny, highly specialized programming language embedded inside python.

Regular Expressions can be used for searching in a string when we don't know exactly what we want to search i.e. it is a special sequence of characters that helps to match or find other strings or set of strings.

Regular Expression patterns: . ^ \$ * ? + + ?
{ } [] \ | () etc.

Regular expressions are also called as REs or regexes or regex patterns.

Module - 4

7a. Explain class and attributes in python language with examples — 05 marks

Answer: class is a user defined type which has a set of variables and methods.

Class is a blue print for creating objects.

A class is an object constructor. Class

is like a factory for creating object.

i.e any number of objects can be created.

Syntax for creating Class

```
Class classname:
    {Statement_1}
    :
    {Statement_2}
```

Annotations: "Class" is labeled as "Keyword", "classname:" is labeled as "user defined name".

Attributes: Attributes are variables

There are two types of attributes

1) Class Attribute

2) Object Attribute

1) class attribute :- class attribute are the variables that are defined directly in the class and those are shared by all objects of the class

We can change or assign the value to the class attribute by using the following syntax

classname.class attribute = value

changing the value at this place will be reflected to all the objects

Syntax to access class attribute or variable is
objectname.classattribute

Q5. Instance attribute :- Attributes defined inside object are specific to object.

Syntax to assign value to instance attribute
objectname.attribute name = value

Syntax to access instance attribute
objectname.attribute name.

Q6. Explain pure functions and modifiers with examples — 05 marks.

Answer :- Pure functions :- Pure function is a function which does not modify any of the object passed as an argument to the user defined function i.e. such functions are called as pure function.

Modifier :- If the user defined function modifies the attributes of the object which is passed as argument of user defined function. Such functions are called as modifiers.

Example; for modifier

class time;

"represents time"

t = time ()

t.hours = 0

t.min = 0

t.sec = 0

```
def increment ( t, second):
```

```
    t.sec + = second
```

```
    if t.sec >= 60:
```

```
        t.sec - = 60
```

```
        t.min + = 1
```

```
    if t.min >= 60:
```

```
        t.min - = 60
```

```
        t.hour + = 1
```

```
increment ( t, 72)
```

```
print ( t.sec)
```

```
print ( t.min)
```

```
print ( t.hour)
```

7c. Write a program that uses class to store the name and marks of students. Use list to store the marks in three subjects - 06 marks

Program

```
class student:
```

```
    def __init__ (self, name, s1=0, s2=0, s3=0)
```

```
        self.name = name
```

```
        self.s1 = s1
```

```
        self.s2 = s2
```

```
        self.s3 = s3
```

```
s = student ("Abhi", 25, 22, 23)
```

```
marks_list = []
```


my

```
marks = list.append(Student("abc", 1, 2, 3))
```

```
for s in marks_list:
```

```
    print(s.name, s.s1, s.s2, s.s3)
```

O/P:

abc 1 2 3

OR

Qa. Explain initialization method with an example — 04 Marks.

Answer: The init method is an initialization method. It is a special method that gets invoked automatically when an object of class is instantiated, i.e. all the attributes inside init method are the object attribute that are instantiated for the class.

The main difference between instance method and init method is instance method is explicitly called whereas init method is not called automatically.

Syntax:

```
def __init__(self, para1, para2, ... paran):  
    Statement
```

Optional parameters

Example

Class Time:

```
def __init__(self, hour = 0, min = 0, second = 0):  
    self.hour = hour  
    self.min = min  
    self.second = second  
  
def print_time(self):  
    print('%0.2d:%0.2d:%0.2d' % (self.hour,  
                                self.min, self.second))
```

t₁ = Time(9, 45, 56)

t₁.print_time()

t₂ = Time(10, 23, 45)

t₂.print_time()

O/P: 09:45:56

10:23:45

Q6 Write a class Rectangle that has attributes length and breadth and a method area which returns the area of the rectangle

06 marks

Answer:

~~Class Rectangle:~~

~~length = 100~~

~~breadth = 200~~

~~area =~~

~~def area_rectangle(l, b):~~

Answer: class Rectangle:

```
def __init__(self, l, b):
```

```
    self.l = l
```

```
    self.b = b
```

```
def area(self):
```

```
    return self.l * self.b
```

```
r = rectangle(10, 10)
```

```
print("area is ", r.area())
```

o/p: area is 100

Q. what is operator overloading? write Python code to overload "+", "-" and "*" operator by providing the methods `--add--`, `--sub--` and `--mul--`

Answer: Ability of an existing operator to work on defined type i.e. class is known as operator.

Python provides a special set of methods which can be used for overloading required operator.

operatorspecial method in python

+

__add__()

-

__sub__()

*

__mul__()

example program

class num:

def __init__(self, x, y):

self.x = x

self.y = y

def __add__(self, p2):

p3 = addition()

p3.x = p1.x + p2.x

p3.y = p1.y + p2.y

return p3

def __sub__(self, p2):

p4 = subtraction()

p4.x = p1.x - p2.x

p4.y = p1.y - p2.y

return p4

def __mul__(self, p2):

p5 = multiplication()

p5.x = p1.x * p2.x

p5.y = p1.y * p2.y

return p5

```
def __str__(self):  
    return "(%.d, %.d)" % (self.x, self.y)
```

```
p1 = num(3, 2)
```

```
p2 = num(5, 4)
```

```
p3 = p1 + p2
```

```
p4 = p1 - p2
```

```
p5 = p1 * p2
```

```
print(p3)
```

```
print(p4)
```

```
print(p5)
```

Module 5

1a. Write a python program for retrieving the someo.txt file from the web and compute the frequency of each word in a file
-06 marks.

Program

```
import urllib.request, urllib.parse, urllib.error  
fhand = urllib.request.urlopen('http://data.pr4e.  
org/romeo.txt')  
counts = dict()  
for line in fhand:  
    words = line.decode().split()  
    for word in words:  
        counts[word] = counts.get(word, 0) + 1  
print(counts)
```

9b. Write a Note on XML

— 05 Marks

Answer: XML looks very similar to HTML, but XML is more structured than HTML

```
<person>
  <name>Chuck</name>
  <phone type = "intl">
    +1 734 303 4456
  </phone>
  <email hide = "yes">
```

```
</person>
```

With reference to above example, an XML document is like a tree structure, where there is a top tag person and other tags such as phone drawn as children of their parent nodes

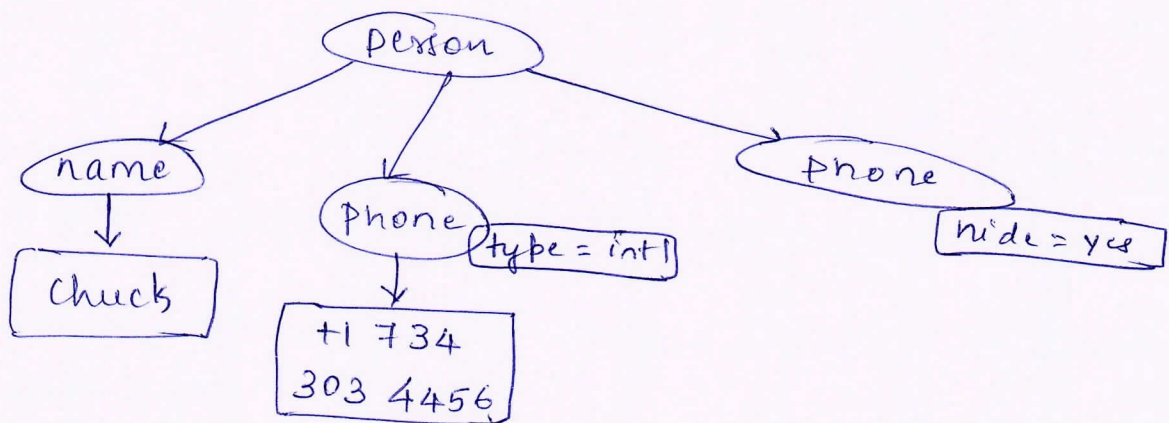


fig: A tree representation of XML.

9c. Explain with a neat diagram of Service Oriented Architecture

Answer: The ability to exchange data between applications using HTTP and a way to represent complex data that we are sending back to forth between these applications using XML or Javascript Object Notation (JSON)

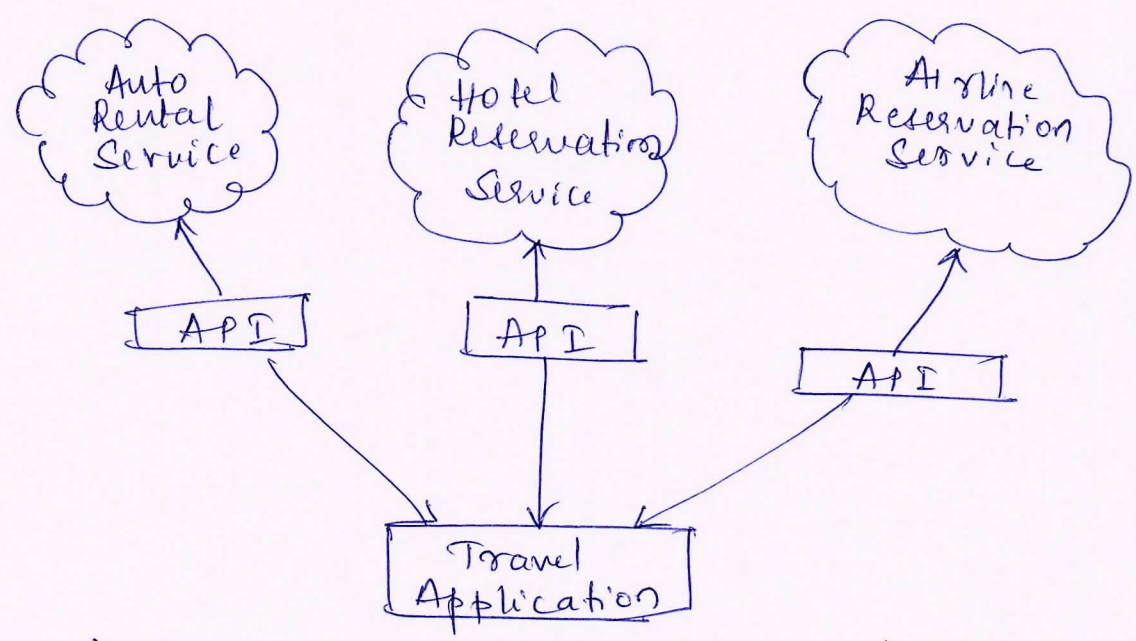


fig: Service oriented Architecture

when we begin name for these application-to-application contracts is API's

when we begin to build our programs where the functionality of our program includes access to services provided by other programs we call approach a service-oriented architecture or SOA

A SOA approach is one where our overall application make use of the services of other application. A non-SOA approach is where

the application is a single standalone application which contains all the code necessary to implement the application.

Service-Oriented Architecture has many advantages including

1) we always maintain only one copy of data

2) owners of the data can set the rules about the use of their data

10a. Describe creation of database table using database cursor architecture — 08 Marks

Answer: Database require more defined structure than python lists or dictionaries.

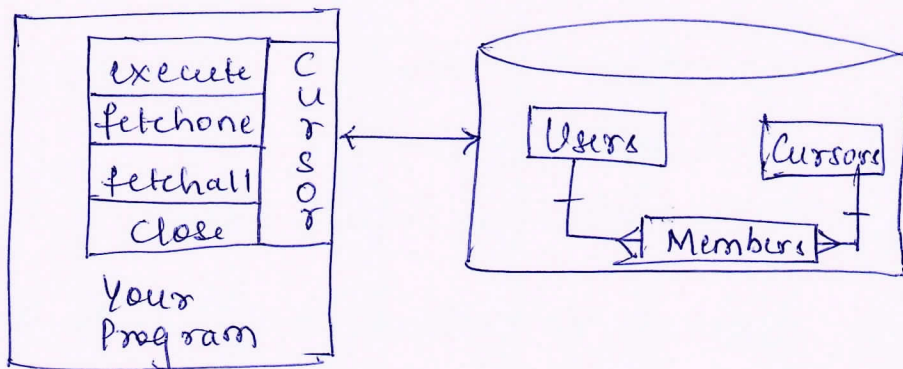


fig: Database Cursor

when we create a database table we must tell the database in advance the names of each column in the table and the type of data which we are planning to store in each column when the database software knows the type of data in each column, it can choose the

most efficient way to store and look up the data based on the type of data.

Defining structure for data up front may seem inconvenient at the beginning, but the payoff is fast access to ~~your~~ data even when the database contains a large amount of data

The code to create a database file and a table named Tracks with two columns in the database is as follows:

```
import sqlite3
conn = sqlite3.connect('music.sqlite')
cur = conn.cursor()
cur.execute('DROP TABLE IF EXISTS Tracks')
cur.execute('CREATE TABLE Tracks (title TEXT, plays INTEGER)')
conn.close()
```

The connect operation makes a "connection" to the database stored in a file music.sqlite3 in the current directory. ~~It~~

Ob. write a python code for creating employee database inserting records and selecting the employees working in the company — 08 marks


```
import sqlite3
```

```
conn = sqlite3.connect('EmployeeDB.db')
```

```
c = conn.cursor()
```

```
c.execute('CREATE TABLE Employee (name, text,  
age Integer, salary float, designation text,  
company text)')
```

```
n = int(input("Enter number of records"))
```

```
for i in range(n):
```

```
    ename = input("Enter name")
```

```
    eage = int(input("Enter age"))
```

```
    esalary = float(input("Enter salary"))
```

```
    edesig = input("Enter designation")
```

```
    ecompany = input("Enter company")
```

```
c.execute("INSERT INTO Employee
```

```
VALUES (?, ?, ?, ?, ?)', (ename, eage,  
    esalary,  
    edesign, ecompany)
```

```
conn.commit()
```

```
c.execute("select * from Employee")
```

```
print(c.fetchall())
```

```
conn.close()
```