KLS Vishwanathrao Deshpande Institute of Technology

(Accredited by NAAC with "A" Grade)
(Approved by AICTE, New Delhi, Affiliated to VTU, Belagavi)
(Recognized Under Section 2(f) by UGC, New Delhi)
Udyog Vidya Nagar, Haliyal - 581 329, Dist.: Uttara Kannada
Phone: 08284 - 220861, 220334, 221409, Fax: 08284 - 220813
www.klsvdit.edu.in | principal@klsvdit.edu.in | hodece@klsvdit.edu.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

University / Model Question Paper Scheme & Solution

Faculty Name	:	Shree Gowri SS
Course Name	:	Python Application and Programming
Course Code	:	18EC646
Year of Question Paper	:	June - July 2023
Date of Submission	:	10/10/2023

Faculty Member

HoD

Head of the Department
Dept. of Electronics & Communication Engg.
KLS V.D.I.T. HALIYAL (U.K.)

Dean (Acad.)

Name of Faculty : Prof. Shree Gowri ss

Name of Institution : KLS VDIT Haliyal

Department : Electronics and Communication Engg.

Sem : 6th sem B.E Degree Examination June/July 2023

Subject : Python Application and Programming

Scheme and Solution

Subject Code: 18EC646

Prof. Shree Gowri ss

Subjet Teacher

HOD

Head of the Department
Dept. of Electronics & Communication Engg.
K.L.S V.D.I.T. HALIYAL (U.K.)

Dean Academic

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGG
KLS VDIT
HALIYAL

CBCS SCHEME

USN | | | | | | | | | | 18EC646

Sixth Semester B.E. Degree Examination, June/July 2023
Python Application Programming

Time: 3 hrs. Max. Marks: 100

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

Module-1

1 a. Explain the concept of type conversion functions and math functions in python with examples. **(10 Marks)**

b. Write a program which prompts the user for a Celsius temperature, convert the temperature to Fahrenheit, and print out the converted temperature. **(05 Marks)**

c. List any five features of python programming language. **(05 Marks)**

OR

2 a. List and give syntax of all python supported conditional statements. Write a python program to check whether given number is positive or negative or zero. **(10 Marks)**

b. Explain the rules of precedence used by python to evaluate an expression. **(05 Marks)**

c. Write a program to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error message. If the score is between 0.0 and 1.0, print a grade using the following table:

Score	>= 0.9	>= 0.8	>= 0.7	>= 0.6	< 0.6
Grade	A	B	C	D	F

(05 Marks)

Module-2

3 a. Write a python code which repeatedly reads numbers until the user enters "done". Once "done" is entered, print out the total, count, and average of the numbers. If the user enters anything other than a number, detect their mistakes using try and except and print an error message and skip to the next number. **(10 Marks)**

b. List and explain with example any five built in string manipulation functions supported by python. **(10 Marks)**

OR

4 a. Define a string. How it can be traversal through using looping statement? **(06 Marks)**

b. Explain file open, file close, file read and file write concepts in python with examples. **(08 Marks)**

c. Write a program to read through a file and print the contents of the file (line by line) all in upper case. **(06 Marks)**

Module-3

5 a. What is dictionary? Write a python program that accepts a sentence and build dictionary with LETTER, DIGIT, UPPER CASE, LOWER CASE as key value and their count in the sentences as values.
Example: Sentence = 'VTU@123.e-Learning'
d = {"LETTER" : 12, "DIGITS" : 3, "UPPER CASE" : 4, "LOWER CASE" :8} **(10 Marks)**

b. Compare and contrast tuples with lists. Explain the following operations in tuples:
(i) Sum of two tuples
(ii) Slicing operators
(iii) Comparison of two tuples
(iv) Assignments to variables **(10 Marks)**

OR

6 a. Describe any two list operations and list methods. Write a python program to accept 'n' numbers from user, find sum of all even numbers and product of all odd numbers in entered list. **(10 Marks)**

b. Illustrate the use of regular expressions for:
(i) Extracting data
(ii) Character matching
(iii) Combining searching and extracting **(10 Marks)**

Module-4

7 a. What is class? How to define a class in python? **(05 Marks)**

b. Write a definition for a class named circle with attributes center and radius, where center is a point object and radius is a number. Instantiate a circle object that represents a circle with its center at (150, 100) and radius 75. Write a function named point_in_circle that takes a circle and a point and returns True if the point lies in or on the boundary of the circle. **(10 Marks)**

c. Distinguish between pure functions and modifiers with example. **(05 Marks)**

OR

8 a. Illustrate how __init__ method is invoked when an object is initiated. **(05 Marks)**

b. What does the keyword self in python mean? Explain with example. **(05 Marks)**

c. What is operator overloading and type-based dispatch? Write a python code to add or increment the time based on the type of second parameter. If the second parameter is time then perform addition. If it is integer then perform increment operation. **(10 Marks)**

Module-5

9 a. What is socket? Explain how socket connection can be established to the internet using python code over the TCP/IP connection and the http protocol to get the web content. **(08 Marks)**

b. Write a note on XML. Write a python program to retrieve a node present in XML. **(08 Marks)**

c. What is service-oriented architecture? List the advantages of the same. **(04 Marks)**

OR

10 a. Brief on structured query language, with suitable python program explain functions involved in creation of database table in python. **(08 Marks)**

b. Demonstrate with the python program:
(i) How to retrieve an image over HTTP?
(ii) How to retrieve web pages with Urllib? **(08 Marks)**

c. Compare and contrast the Java Script object notation and extensible markup language (JSON and XML). **(04 Marks)**

* * * * *

Module - 1

1.a :- Explain the concept of type conversion and math function in python with example — 10 marks

scheme : concept of type conversion — 05 marks
math function in python — 05 marks
10 marks

Solution :- Type conversion.

values in the python can be converted from one type to another, this is called as type casting.

i> int/str to float

Ex :- (1)

```
>>> x = 5
>>> float(x)
>>> print(x)
```

o/p 5.0

* float() is function is used to convert int to float & str to float

ii>

Ex :- (2)

```
>>> x = "5"
>>> float(x)
>>> print(x)
```

o/p 5.0

```
>>> type(x)
```

o/p <class 'float'>

ii> float/string to int

Ex :- (1)

```
x = 5.0
int(x)
print(x)
```

o/p > 5

```
type(x)
```

<class 'int'>

(2)

```
x = "5"
int(x)
print(x)
```

o/p > 5

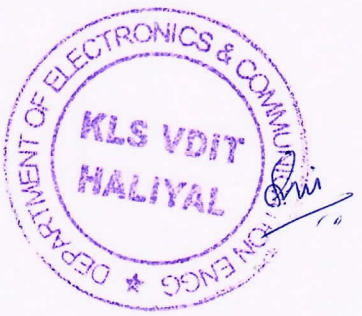
Head of the Department
Dept. of Electronic & Communication Engg.
KLS V.D.I.T., HALIYAL (U.K.)

Python has a math module that provides most of the familiar mathematical function. Before we can use the module, we have to import it

Ex:-
```
import math          # import math module
math.sin(90*)        module name . function name
math.log10(x)
x = math.pi
```

1b. Write a program the users for a Celsius temperature, convert the temperature to fahrenheit & print out the converted temperature — 05 marks

Sol:-
```
inp = input(" Enter Celsius temperature:")
Cel = float(inp)
fahr = 32.0 + Cel * 9.0/5.0
print (" temperature in fahrenheit")
print (fahr)
```



1c. List any five features of ~~the~~ python language — 5 marks

1 x 5 = 5

Sol:- five features of python language are:-

1. Python is high-level language
2. Both procedure oriented and object-oriented language.
3. It is an interpreted language
4. Easy learn & easy to use
5. Rich in library.

2a. Explain conditional execution, Alternative execution, chained conditionals and Nested conditionals with examples — 10 marks

5 x 02 = 10

Sol :- i) Conditional Execution

Syntax

```
if condition:
    Statement
```

Ex :- Number is positive or negative

```
if num > 0:
    print ("The number is positive")
```

ii) Alternative Execution

Syntax :

```
if condition:
    Statement_A
else:
    Statement_B
```

Ex :-

```
if num > 0:
    print ("The number is positive")
else:
    print ("The number is negative")
```

iii) chained conditionals

Syntax :-

```
if condition:
    Statement_1
elif condition:
    Statement_2
else:
    Statement_3
```

Note :- In chained conditionals any number of elif conditions can be specified and not mandatory to specify else or else is optional.

Ex :-

```
if num > 0:
    print ("The number is positive")
elif num < 0:
    print ("The number is negative")
else:
    print ("The number is zero")
```

iv) Nested conditionals

It is Nesting of if - else in either if block or in else block.

Syntax :-

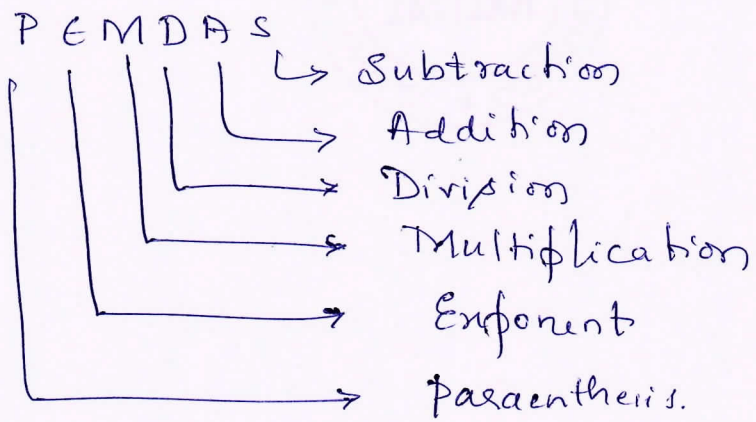
```
if condition1:
    if condition1:
        Statement_1
    else:
        Statement_2
else:
    Statement_n
```

Example :-

```
if num >= 0:
    if num > 0:
        print("No is positive")
    else:
        print("Number is zero")
else:
    print("Number is Negative")
```

2b. Explain the rules of precedence used by Python to evaluate the expression. — 04 Marks

Sol:- The rules of precedence used by Python to evaluate the expression is PEMDAS



Paraenthesis is having highest preference

Exponent is having next level of priority.

Multiplication & Division are at the same priority level

Addition & Subtraction remains at same level of priority.

Ex :- $2 * (3^2 - 2^2)$

In the above example the parenthesis is evaluated first. $(3-4) = 5$ $\therefore 2 * 5 = 10$.

1c. Write a python program to prompt the user for hours and rate per hour for pay computation with time and half for overtime.

2c. Write a python program to prompts for a score between 0.0 and 1.0. If the score is out of range, print an error message. If the score is between 0.0 and 1.0 print a grade using the following table

Score	>=0.9	>=0.8	>=0.7	>=0.6	<0.6
Grade	A	B	C	D	F

— 05 marks

Sol:-

```
score = float (input ("enter the score: "))
if score < 0.0 or score > 1.0:
    print ("The score is out of range")
elif score >= 0.9:
    print ("The grade is A")
elif score >= 0.8:
    print ("The grade is B")
elif score >= 0.7:
    print ("The grade is C")
elif score >= 0.6:
    print ("The grade is D")
elif score < 0.6:
    print ("The grade is F")
```

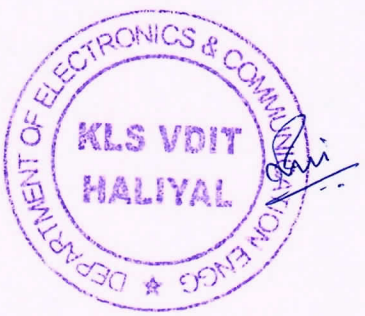
Module - 2

3a. Write a python code which repeatedly reads numbers until the user enters "done". Once "done" is entered, print out the total, count, and average of numbers. If the user enters anything other than a number, detect their mistake using try and except and print out an error message and skip to the next number — 10 Marks

Sol:-

```
total = 0
count = 0
avg = 0
try:
    while (true):
        num = input("enter the number : ")
        if num == "done":
            break
        total = total + num
        count = count + 1
        avg = total / count
except:
    print("total :", total)
    print("count :", count)
    print("avg :", avg)

except :
    print("Please enter valid input number")
```



b. List and explain with example any five built in string manipulation functions supported by python — 10 Marks

Sol:-

1) capitalize() :- This method converts first characters to upper case

Ex:- txt = "hello"

x = txt.capitalize()

print (x)

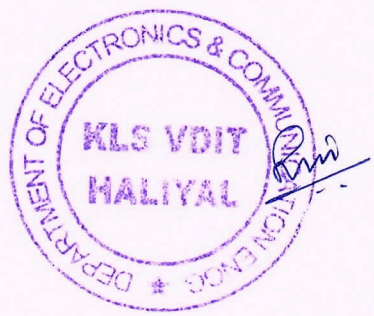
O/P : Hello

2) islower() :- Returns True if all characters in the string are lower case

Ex:- txt = "hello world"

x = txt.islower()

print (x)



O/P : True

3) lower() : This method converts all characters to lower case

Ex:- txt = "Hello World"

x = txt.lower()

print (x)

O/P : hello world

4) upper() :- This method converts all the characters to upper case

Ex:- txt = "hello world"

x = txt.upper()

print (x)

O/P : HELLO WORLD

OR

4a. Define string. How it can be traversal through. using looping Statement — 06 Marks

Sol:- String is a sequence of character, we can access the characters one at a time with the bracket operator. — 01 M

Ex:- fruit = "banana"

letter = fruit[1]

A lot of computation involves processing a string one character at a time.

There are two loops to traverse the string — 05 M

06 M

i) using while loop

Ex:-

```
index = 0
fruit = "banana"
while index < len(fruit):
    letter = fruit[index]
    print(letter)
    index = index + 1
```


ii) using for loop

```
for i in fruit:
    letter = fruit[i]
    print(letter)
```

4b:- Explain the file open, file close, file read & file write concepts in python with examples - 08 mark

Sol:- opening file

open() function is used to open the file.

Syntax:- open (filename, mode)

Ex:- handle = open ('filename.txt', mode)

There are four modes for opening the file

1) r - read mode, if mode not specified in open() then by default it is readmode

2) w - open the file for the purpose of write operation

3) a - append → opens file for appending

4) x - creates the specified file

Ex:- fhand = open ('mbox.txt')
print (fhand)

Reading file

The content of the file can be read, first open the file, which returns the file handle also known as objech

Ex:- fhand = open ("filename.txt")
print (fhand.read())

Writing file :- In order to write information in a file write() method is used with the file handler After writing the file, the file must be closed using close() method

Ex:- fhand = open ("filename.txt")

```
line = "Welcome"
fhand.write(line)
fhand.close()
```

4c. Write a program to read through a file and print the contents of file (line by line) all in upper case — 06 marks

Sol:-

```
fname = input("Enter the filename:")
try:
    fhand = open(fname)
except:
    print("file cannot be opened", fname)
    exit()
for line in fhand:
    line.upper()
    print(line)
```

Module - 3

5a. What is dictionary? Write a python program that accepts a sentence and build dictionary with LETTER, DIGIT, UPPERCASE, LOWERCASE as key value and their count in the sentences as values.

Example: Sentence = "VTU@123. e-Learning"

d = {"Letter": 12, "Digits": 3, "Uppercase": 4, "lowercase": 8} — 10 Marks

Sol: Dictionary: is a mapping between set of indices (keys) and set of values. It is denoted by { }

{key : value}

Pwgm:-

```
d = { }
sentence = input("enter a sentence: ")
d['letters'] = 0
d['Digits'] = 0
d['uppercase'] = 0
d['lower case'] = 0
for c in sentence:
    if (c >= 'a' and c <= 'z')
        d['lowercase'] = d['lowercase'] + 1
        d['letters'] = d['letters'] + 1
    if (c >= 'A' and c <= 'Z')
        d['Uppercase'] = d['uppercase'] + 1
        d['letters'] = d['letters'] + 1
    if (c.isdigit())
print d['Digits'] = d['Digits'] + 1
```

5b. Compare and constrat tuples with lists. Explain the following operation in tuples. — 10 marks

(i) Sum of two tuples
(ii) Slicing operators
(iii) Comparison of two tuples
(iv) Assignments to variables

Sol:- Comparision — 02 marks
operation of tuple - 04 x 2 = 08 marks

The difference between list and tuple are

List	Tuple
1. Elements in the list are enclosed in square bracket [ ]	1. Elements in the tuple are enclosed in ( )
2. list are mutable	2. Tuple are immutable.

(i) Sum of two tuples

$t_1 = (1, 2, 3)$

$t_2 = (a, b, c)$

$t = t_1 + t_2$

$= (1, 2, 3, a, b, c)$

(ii) Slicing operations

Ex:- t = ('a', 'b', 'c', 'd', 'e')

print (t[1:3])

O/P ('b', 'c')

iii) Comparison of two tuples

```
>>> t1 = (1, 2, 3)
>>> t2 = (1, 2, 3)
>>> t3 = t1 == t2
>>> print (t3)        O/P: True
```

iv) Assignments to variables

Ex:- fruits = ('mango', 'orange', 'cherry')

```
yellow, green, red = fruits
print (yellow)
print (green)
print (red)
```

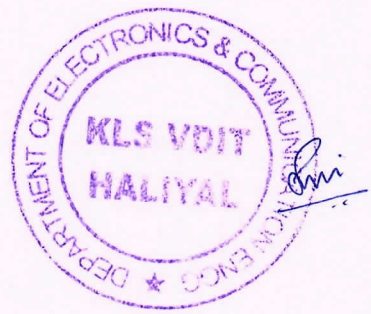
OR

6a. Describe any two list operations and its methods. write a python program to accept 'n' numbers from user find sum of all even numbers and product of all odd numbers in entered list — 10 marks

Sol:- List operations

1) The + operator:- It concatenates the list

```
>>> a = [1, 2, 3]
>>> b = [4, 5, 6]
>>> c = a + b
>>> print (c)
o/p: [1, 2, 3, 4, 5, 6]
```



2) * operator :- This operator is used to create a repeated sequence of list items

```
>>> a = [1, 2, 3] * 3
>>> print (a)
o/p [1, 2, 3, 1, 2, 3, 1, 2, 3]
```

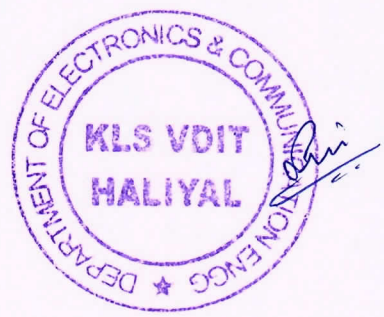
List methods

1) append() :- Adds new element to the end of the list

Ex:-
```
t = ['a', 'b', 'c', 'd']
t.append('e')
print(t)
```
o/p ['a', 'b', 'c', 'd', 'e']

2) clear() :- clears the list

```
t = ['a', 'b', 'c', 'd', 'e']
t.clear()
print(t)
```
o/p []




```
import math
print("Enter the number & type done to complete:")
evenlist = []
oddlist = []
while True:
    inp = input()
    if inp == "done":
        break
    elif in(inp) % 2 == 0:
        evenlist.append(int(inp))
    else:
        oddlist.append(int(inp))
sumeven = sum(evenlist)
productodd = math.prod(oddlist)
print("Even list are: {}", format(evenlist))
print("odd list are: {}", format(oddlist))
```

print ("Sum of even numbers are: {}", format(sumeven))
print ("Sum of odd numbers are: {}", format(productodd))

6b. Illustrate the use of regular expressions for — 10 marks

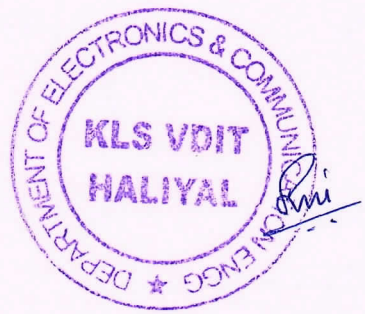
(i) Extracting data — 03 M

(ii) character matching — 03 M

(iii) Combining searching & extracting — 04 M

Sol ii) Character matching in regular expression

```
import re
hand = open ("filename.txt")
for line in hand:
    line = line.rstrip()
    if re.search ('h', line):
        print (line)
```



i) Extracting data using RE

Ex:-
```
import re
s = "A message from csev@umich.edu to
     cwen@iupi.edu about meeting @ 2PM"
lst = re.findall ('\S+@\S+', s)
print (lst)
```

o/p : csev@umich.edu

iii) Combining searching & extracting

```
import re
hand = open ('mbox-short.txt')
for line in hand:
    line = line.rstrip()
    if re.search ('^X-\S*: [0-9.]+', line):
        print (line)
```

Module - 4

7a. What is class? How to define a class in Python? — 05 M

Sol:- Class is a user-defined type. It is a blue print for creating objects. A class is an object constructor class is a prototype — 02 M

Syntax to create class — 02 M

```
Class classname:
    <statement_1>
    .
    .
    .
    <statement_2>
```

Ex:- — 01 M

```
Class point:
    pass
```

7b. Write a definition for a class named circle with attributes center and radius, where center is a point object and radius is a number. Instantiate a circle object that represents a circle with center at (150, 100) and radius 75. Write a function named point_in_circle that takes a circle and a point and returns True if the point lies in or on the boundary of the circle — 10 Marks

Sol:-

```
import math
Class point:
    " Represents a point attributes: x, y'

Class circle:
    " Represent a circle, Attributes: center, radius"
circle = circle()
Circle . center = point()
Circle . center . x = 150
Circle . center . y = 100
Circle . radius = 75
```

```
def distance_between_point(P1, P2):
    distance = math.sqrt((P2.x - P1.x)**2 + (P2.y - P1.y)**2)
    return distance

def point_in_circle(point, circle):
    d = distance_between_point(point, circle.center)
    return d <= circle.radius

point = Point()
point.x = 175
point.y = 125
print(point_in_circle(point, circle))
```

7c. Difference b/wn pure function and modifiers with example 05m

Sol:- The difference b/wn Pure function and Modifiers are:-

Pure function	Modifier
1. A function that does not modify any of the object it receives as argument.	1. A function that changes one or more of the objects it receives as arguments
2. Pure functions are fruitful	2. Modifiers are not that fruitful
3. A pure functions are has no effect i.e just return a value	3. Changes are visible to caller

Head of the Department
Dept. of Electronic & Communication Engg.
KLS V.D.I.T. HALIYAL (U.K.)

8a. Illustrate how __init__ method is invoked when an object is initiated — 05 marks

Sol:- __init__ method is an initilization method. It is a special method that gets invoked automatically when an object of a class is instantiated. i.e all the attributes of the object instantiated.

Syntax:-

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

(optional parameters)

Ex:-

```
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 ('%.2d:%.2d:%.2d' %(self.hour, self.min,
                                  self.second))

t1 = Time (9, 45, 56)
t1.print_time()
t2 = Time (10, 23, 45)
t2.print_time()
```


When two objects are created, automatically __init__ method is invoked, the attributes of __init__ method are going to become attributes of time object.

8b. What does the keyword Self in python mean? Explain with example — 05 marks

Sol:- Self is a parameter of a method. Self is used to access attributes and methods of the class in python. Self is always pointing to the current object.

Ex:-
```
class point:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y
    def __add__(self, p2):
        p3 = point()
        p3.x = self.x + p2.x
        p3.y = self.y + p2.y
        return p3
    def __str__(self):
        return "(%.d, %.d)" % (self.x, self.y)
p1 = point(2, 3)
p2 = point(4, 5)
print("P1 is :", p1)
print("P2 is :", p2)
p4 = p1 + p2
print("Sum is :", p4)
```

p1 is object of class point
when p1 object is created automatically __init__ method is invoked. when __add__ method is invoked due to +, then self points to current i.e p1 and other is p2.

8c. What is operator overloading and type-based dispatch? Write a python code to add or increment the time based on the type of second parameter. If the second parameter is time it perform addition. If it is integer then perform increment operation — 10 Marks

Sol:- Operator overloading :- Ability of an existing operator to works on programmer defined type i.e Class is known as operator overloading — 01 mark

Type-Base dispatch :- It dispatches the computation to different methods based on type of argument — 01 mark

Ex:-

```
class Time:
    def __init__(self, h=0, m=0, s=0):
        self.hour = h
        self.min = m
        self.sec = s
    def time_to_int(self):
        minute = self.hour * 60 + self.min
        seconds = minute * 60 + self.sec
        return seconds
    def int_to_time(self, seconds):
        t = Time()
        minutes, t.sec = divmod(seconds, 60)
        t.hour, t.min = divmod(minutes, 60)
        return t
```

```
def __str__(self):
    return "%.2d:%.2d:%.2d" %(self.hour, self.
                              min, self.sec)

def __add__(self, t)
    if isinstance(t, Time):
        return self.addTime(t)
    else:
        return self.increment(t)

def addTime(self, t):
    seconds = self.time_to_int() + t.time_to_int()
    return self.int_to_time(seconds)

def increment(self, seconds):
    seconds += self.time_to_int()
    return self.int_to_time(seconds)

T1 = Time(3,40)
T2 = Time(5, 45)
print("T1 is:", T1)
print("T2 is:", T2)

T3 = T1 + T2
print("T1+T2 is:", T3)

T4 = T1+75
print("T1+75 is:", T4)
```

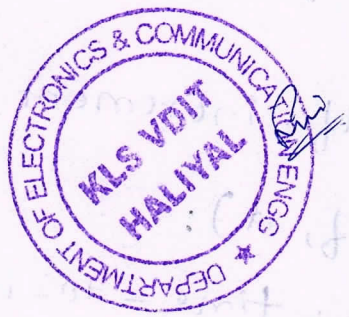
O/P

T_1 is : 03 : 40 : 00

T_2 is : 05 : 45 : 00

$T_1 + T_2$ is : 09 : 25 : 00

$T_1 + 75$ is : 03 : 41 : 15



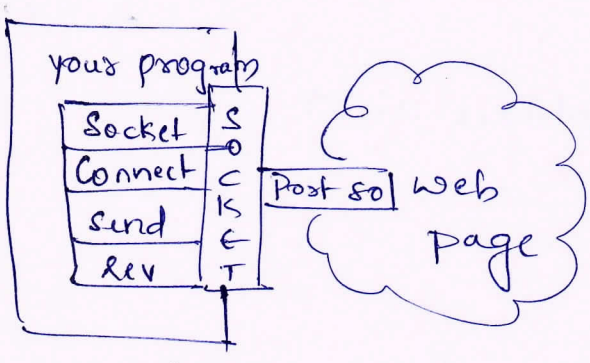
Module - 5

9a. What is socket? Explain how socket connection can be established to the internet using python code over the TCP/IP connection and the http protocol to get the web content (08 marks)

Sol: Socket is a built in support in python, used to make network connection and retrieve data over it in a program — 1 Mark

Program

```
import socket
mysock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
mysock.connect(("data.pr4e.org", 80))
cmd = 'GET http://data.pr4e.org/romeo.txt HTTP/1.0
                                    \r\n\r\n'
mysock.send(cmd)
while True:
    data = mysock.recv(512)
    if len(data)<1:
        break
    print(data.decode(), end="")
mysock.close()
```

Socket connection

Program makes a connection to port 80 on the server : www.py4e.com. The role of program is "web browser". the HTTP protocol says we must send the GET command followed by blank line.

\r\n signifies an EOL.

The program is written with a loop that receives data in 512 - character chunks from the socket and prints out the data until there is no more to read

Q6. Write a note on XML. Write a python program to retrieve a node present in XML. (08 Marks)

Sol :- XML - Extensible markup language

It is one of format used to exchange data across the web — 1M.

Ex :- <Person>

<name> Chuck </name>

<phone type = "intl">

+1734 3034456

</phone>

<email hede = "yes" />

</person>

Program — 07M

```
import xml.etree.ElementTree as ET
input = '''
<stuff>
    <users>
    <user x = "2">
        <id> 001 </id>
        <name> Chuck </name>
    </users>
```

```
</users>
</stuff>'''

stuff = ET.fromstring(input)
lst = stuff.findall('users/user')
print('User count:', len(lst))

for item in lst:
    print('Name', item.find('name').text)
    print('Id', item.find('id').text)
    print('Attribute', item.get('x'))
```

9c. What is service-oriented Architecture? List the advantages of the same. 04 marks

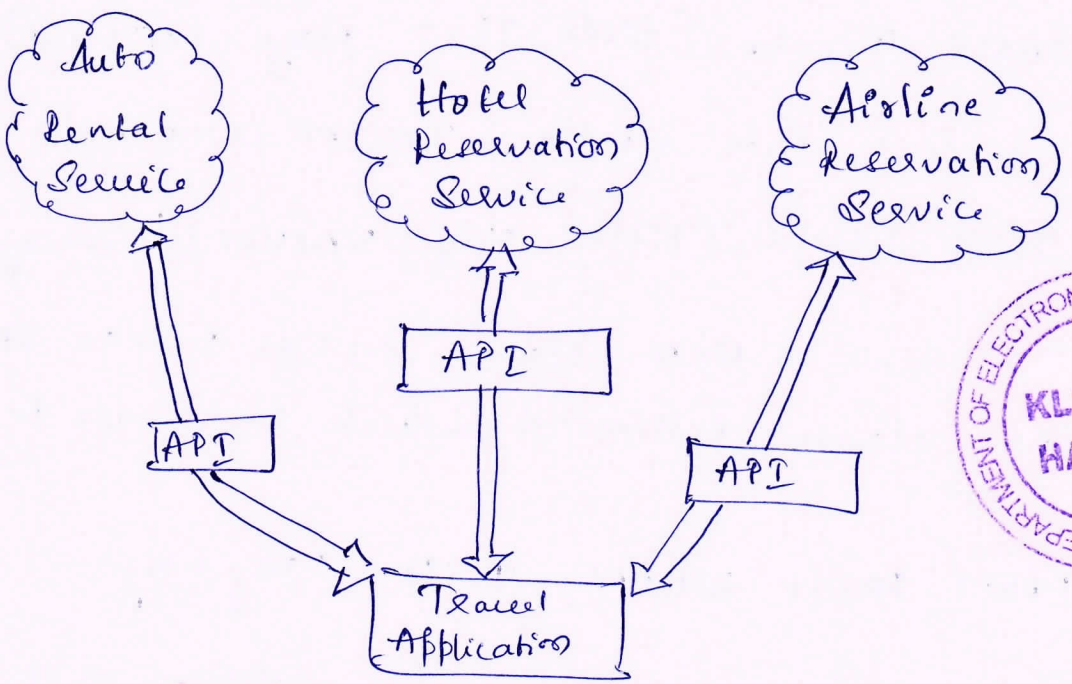


fig: Service Oriented Architecture

- SOA approach is one where our overall application makes use of the services of other Applications. A non SOA approach is where the application is a single stand alone application which contains all of the code necessary to implement the application.

Service oriented architecture has many advantages

(1) we always maintain only one copy of data

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

when an application makes a set of services in its API available over the web, we call these web services.

10 a. Brief on structure query language, with suitable python program explain functions involved in creation of database table in python - 08 marks

Sol:- SQL was standardized so we could communicate in a portable manner to database systems from multiple vendors.

Function involved in creation of database table is

```
CREATE TABLE Tracks (title TEXT, plays INTEGER)
```

To insert a row into a table, INSERT command is used

```
INSERT INTO Tracks (title, plays) VALUES ('My way', 15)
```

To remove a row, WHERE clause on SQL DELETE statement. The WHERE clause determines which rows are to be deleted.

```
DELETE FROM Tracks WHERE title = 'My way'
```

To update a column or columns within one or more rows in a table using SQL UPDATE

```
UPDATE Tracks SET plays = 16 WHERE title = "My way".
```

The four basic SQL commands INSERT, SELECT, UPDATE, DELETE

10.b Demonstrate with a Python program.
(i) How to retrieve an image over HTTP
(ii) How to retrieve web pages with urllib

Sol:- — 08 Marks

```
import socket
import time
Host = "data.pr4e.org"
Port = 80
mysocks = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
mysock.connect((Host, PORT))
mysock.sendall(b'GET http://data.pr4e.org/cover3.jpg HTTP/1.0\r\n\r\n')
count = 0
picture = b""
while True:
    data = mysock.recv(5120)
    if len(data) < 1:
        break
    count = count + len(data)
    print(len(data), count)
    pictures = pictures + data
mysock.close()
pos = picture.find(b"\r\n\r\n")
print("Header length", pos)
print(picture[:pos].decode())
picture = picture[pos+4:]
fhand = open("stuff.jpg", "wb")
fhand.write(picture)
fhand.close()
```

ii) Retrieving web pages with urllib

```
import urllib.request, urllib.parse, urllib.error
fhand = urllib.request.urlopen("http://data.pr4e.
                                org/romeo.txt")
for line in fhand:
    print(line.decode().strip())
```

10 C. Compare and contrast JSON and XML -04 marks

Sol:- The difference between JSON and XML.

JSON	XML
1. JSON uses Key-value pair	1. XML uses tag.
2. Simpler	2. complicated