CBCS SCHEME

USN | | | | | | | | | | |

15CS664

Sixth Semester B.E. Degree Examination, Jan./Feb. 2021
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 the computer hardware architecture with a neat sketch. (06 Marks)
b. Write a note on general types of errors. (06 Marks)
c. Write a program that uses input to prompt a user for their name and then welcomes them. (04 Marks)

OR

2 a. Write a program which prompts the user for a Celsius temperature, convert the temperature to Fahrenheit and print the converted temperature. (06 Marks)
b. Explain nested conditional statement with an example. (04 Marks)
c. Write a program with a function computer grade that takes a score as its parameter and returns a grade as a string. (06 Marks)

Module-2

3 a. Analyze the use of break and continue statement with an example. (06 Marks)
b. Explain format operators in python with suitable examples. (03 Marks)
c. Define a file data structure. Illustrate reading and writing operation on files with examples. (07 Marks)

OR

4 a. Write a program to read numbers repeatedly until the user enters 'done'. Once 'done' is entered print out total, count and average of the numbers. (06 Marks)
b. Write a note on string methods. (07 Marks)
c. Write a program to read through a file and print the contents of the file (line by line) all in upper case. (03 Marks)

Module-3

5 a. Explain list operations and list methods with examples. (05 Marks)
b. Write a program to count how many times each letter appears in a word. (07 Marks)
c. Explain tuple assignment with examples. (04 Marks)

OR

6 a. Write a program to open a file and read it line by line. For each line, split the line into list of words using split function. For each word check to see if the word is already in a list. If the word is not in the list, add it to the list. (06 Marks)
b. Explain advanced text parsing using dictionary. (07 Marks)
c. Why search and find all functions of regular expressions used? Explain with suitable examples. (03 Marks)

Module-4

7 a. Define class. Explain classes and objects of python in detail with suitable examples. (10 Marks)

b. What is a pure function? Explain with an example. (06 Marks)

OR

8 a. Write a program with a function print_time that takes a time object and prints it in the form hour:minute:second. Write another function is_after that takes two time objects t_1 and t_2 and resume True if t_1 follows t_2 chronologically and False otherwise. (08 Marks)

b. Write a note on operator overloading with an example. (08 Marks)

Module-5

9 a. How to retrieve web pages using urllib? Explain how to compute frequency of each word in the file retrieved. (08 Marks)

b. What is an API? Explain with a neat sketch. (08 Marks)

OR

10 a. Write a program to read binary files. (08 Marks)

b. Explain keys in a database model. (08 Marks)

* * * * *

Name of faculty : Prof. Ravindra Patil

Department : Computer Science & Engg.

Sem : 7th sem.

Subject : Python Application Programming.

Subject code : 15CS664/17CS664/18CS752.

Staff Incharge

HOD

Computer Science & Engineering
KLS Vishwanathrao Deshpande
Institute of Technology, Haliyal.

Dean Academic

Dean, Academics
KLS VDIT, HALIYAL

Module - 1

1a. Explain the computer hardware architecture with a neat sketch — 06 marks

Answer: The basic architecture of computer hardware

The important parts of the computer are

1) Central Processing Unit :- It performs basic arithmetic & logical as well as control of I/O operations. The speed of computer is tremendous, suppose if computer is rated at 3.0 Gigahertz it means CPU will execute 3 billion instructions per second.

2) Main Memory: It is a storage area. Any program or task that need to be executed by CPU must be in main memory. Main memory is nearly as fast as CPU. But the information stored in main memory vanishes when the computer is turned off.

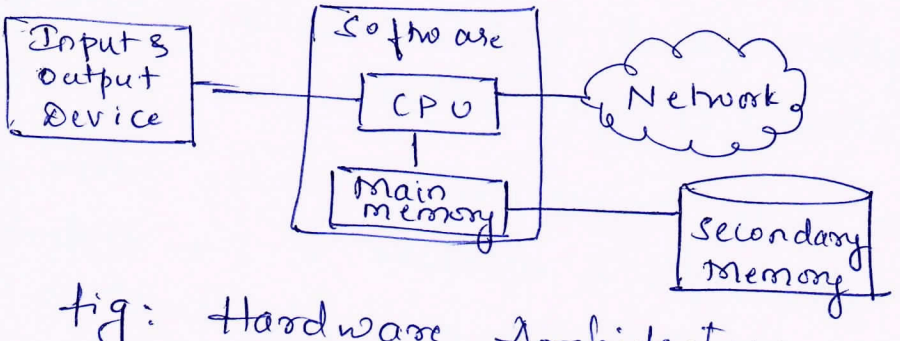

fig: Hardware Architecture

3) Secondary Memory : Secondary memory is also to store information but much slower than main memory. The advantage of secondary memory is that it can store the information even when there is no power to computer

Examples of secondary memory are :- disk drives, flash memory, USB etc

4) Input and Output devices: I/O devices act as communication medium b/w end user & a computer examples of I/O devices are keyboard, mouse, microphone, printer, speaker, touchpad etc

5) Network connection :- Most of computer are connected to a network & hence they can communicate with other computers in a network i.e we can retrieve information over a network. Network is very slow place to store & retriev the data i.e it is slower than secondary memory.

1b. Write a note on general ~~purpose~~ types of errors

Answer The three general types of errors are

1) Syntax error: The statements that are not following the grammar of the programming language. Python point right at the line & character where it noticed it got confused

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 the statement in not proper.

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

1c. Write a program that uses input to prompt a user for their name and then welcomes them (04 marks)

Answer To write python program it requires two funcns
1) input() 2) print()

```
>>> name = input("Enter your name: \n")
>>> print("welcome ", name)
```

O/P Enter your name
xyz
welcome xyz

2a Write a program which prompts the user for a Celsius temperature, & convert the temperature to Fahrenheit & print the converted temperature. (06 Marks)

Answer: 1. Formulas for converting Celsius to temperature

$F = (°C \times 1.8) + 32$

2. Read the Celsius temperature
3. Convert the temperature to Fahrenheit
4. print the converted temperature

```
>>> Celsius = float(input(" Enter the Celsius temperature: "))
>>> f = (Celsius * 1.8) + 32
>>> print("Fahrenheit temperature: ", f)
```

O/P Enter the celsius temperature: 23.5
Fahrenheit temperature: 74.3

2b. Explain nested conditional statement with an example (04 marks)

Answer One condition can also be nested inside another. i-e an if statement contains another if statement either if block or in else block then it is called as Nested statement

Syntax of Nested conditional statement

```
if boolean_exp_1 :
    if boolean_exp_a :
        ~~Stat~~Statement_1
    else :
        Statement_2
else :
    Statement_last
```

If the boolean_exp_1 is evaluated to True the control is transferred to boolean_exp_2 & it is evaluted. If it evaluates to True statement_1 is executed else statement_2 is executed.

If boolean_exp_1 is evaluated to False then statement_last is executed.

2c. Write a program with a function computer grade that takes score as its parameter and returns a grade as a string (06 marks)

Answer: * Creati a user defined function called computer grade which takes score as parameter

* function should return Grade as a string.

* let us make an assumption

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

* To create user defined function use def keyword & name of function is computer grade & parameter passed is ~~Grade~~ Score.

program

```
>>> def computer grade (score):
        if score > 1.0 or score < 0.0:
            print (" Score is out of range")

        elif score >= 0.9:
            print(" Grade is A")

        elif score >= 0.8:
            print ("Grade is B")

        elif score >= 0.7:
            print (" Grade is C")
        elif score >= 0.6:
            print (" Grade is D")
```

```
elif score < 0.6 :
        print [" Grade is F")
>>> x = float(input(" Enter the Score: "))
>>> computergrade(x)
```

Module 2

3a Analyze the use of break and continue statement with an example (06 Marks)

Answer

break statement: break is a keyword in python break statement is useful in case of loops ~~statements~~ i.e when loop is going to become an Infinite loop. In order to come out from the Infinite loop break statement is used.

consider while loop

```
>>> n=1
>>> while True:
        print(n)
        n=n+1
```

In the above example program while loop becomes True forever & it continues execute Therefore to come out from the loop break statement is used

Syntax:

```
while testexpression:
    if condition:
        break
```

example

```
while True:
    line = input( )
    if line == 'done':
        break
    print (line)
```

i.e if condition in while loop becomes true then break statement is executed & it comes out of the while loop.

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

3b. Explain format operators in python with suitable examples (03 marks)

Answer: % is the format operator
This format operator replaces parts of the string with value stored in the variable.

The format sequences are

1) % d → Integer
2) % g → floating point numbers
3) % s → for string.

Example:

```
age = 25
print (" my age is % d " % age)
```

format sequence ↓ variable ↓

o/p: my age is 25

3c. Define a file data structure. Illustrate reading and writing operation on files with examples (07 Marks)

Answer File is a common storage unit in a computer. It is used to store the information. The operations that can be performed on file are

1) read
2) open
3) write
4) close.

Syntax to open file

open (filename , mode)

filename → It is parameter which indicates the name of the file along with the path.

mode → We can open the file for different purpose like for reading, writing etc

default mode is read mode.

read mode → 'r'

write mode → 'w' etc

1) Reading operation on files.

The file can be open for reading the content of the file. The mode is 'r'

if mode not specified then automatically default is read mode.

file_handler = open (filename, mode)

Example

handle = open ("filename.txt", "r")

or

handle = open ("filename.txt")

4a write a program to read numbers repeatedly until the user enters 'done'. Once 'done' is entered print out total, count and average of the number.

Answer

```
n = 0
count = 0
total = 0
average = 0
while True:
    try:
        n = input(" enter a number: ")
        if n == "done":
            break
        n = int(n)
        count = count + 1
        total = total + n
        average = total / count
    except:
        print(" Enter a valid number")
print(" count : ", count)
print(" total : ", total)
print(" average: ", average)
```

To read the content of file read() function is used

```
handle = open("filename.txt", "r")
print (handle.read())
```

Illy 1) we can count the number of lines in file by using for loop.

2) we can count number of words in file by using len() function etc.

2) <u>writing files</u>

To write content to file, then open the file in w mode i.e write mode. If the file we are opening to write already exists then earlier content is ~~cancel~~. ~~eras~~ cleared.

<u>Example</u>

```
handle = open("filename.txt", "w")
line1 = " hi how ru"
handle.write(line1)
handle.close()
```

after writing the contents into the file, the file needs to be closed. This is done in order to save the contents.

4b: Write a note on string methods (07 marks)

A method is similar to function it takes argument & return value

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

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

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

ex:-
```
>>> "Exam".lower()
o/p exam
```

3) capitalize() :- This method returns a string with its first character capitalized & remaining are lower case

ex:-
```
Quote = " Hi how Ru you"
Quote.capitalize()
Hi how ru you.
```

4) find() :- checks if substring appears in string-name & 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 occurance of substring in string.

```
>>> language = "Monty Python"
>>> language.count("n")
2
```

6) replace() :- This method replaces all occurence of old in string with new. If the optional argument max is given, then only the first few max occurance are replaced.

4c. Write a program to read through a file & prints the contents of the file (line by line) all in upper case. (03 marks)

Answer:

```
fhand = open("filename.txt", "r")
for line in fhand:
    b = line.upper()
    Print(b)
```

Module 3

5a Explain list Operations and list methods with examples (05 marks)

The list operations are + and *

The + operator is used to concatenate the list

```
>>> a = [1, 2, 3]
>>> b = [4, 5, 6]
>>> c = a+b
```

print (c)

Op: [1, 2, 3, 4, 5, 6]

The * operator: This operator is used to create a repeated sequence of list items

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

<u>in operator</u> :- This operator checks the item in the list

```
>>> character = ["abc", "def", "xyz"]
>>> "pqr" in character
False
```

<u>Methods in list</u>

1) append () : Adds an element at the end of the list

2) clear () :- Removes all the elements from the list.

3) copy () :- returns a copy of the list

4) pop () :- removes element at the specified position from the list

5) sort () :- sort the list in ascending order.

5b. Write a program to count how many times each letter appears in a word (07 marks)

Answer Program : 1> count letter in word
2> word is fruit
3> Use dictionary which holds character as key & count as value

```
word = "fruit"
d = dict()          # d = { }
for char in word:
    print(" character: ", char)
    if char not in d:
        d[char] = 1
    else:
        d[char] = d[char] + 1
print("d = ", d, "\n")
```

5C. Explain tuple assignment with examples (04 marks)

Answer: Tuple has a unique feature of assigning values of tuple to multiple variables at a time using assignment statement, where at the LHS of assignment operator tuple variables are specified & at the RHS of assignment operator we have tuple values. This is also called as unpacking tuple.

```
fruits = ("apple, "banana", "cherry")
green, yellow, red = fruits
print (green)
print (yellow)
print (red)
```

o/p: apple
banana
cherry

The number of variables on the left & the number of values on the right must be same.

6a Write a program to open a file and read it line by line. For each line, split the line into list of words using split function. For each word check to see if the word is already in a list. If the word is not in the list, add it to the list. (06 Marks)

Answer:

```
fname = input (" Enter the filename : ")
try:
    fhand = open(fname)
except:
    print(" file cannot be opened)
    exit()
d = dict()
for line in fhand:
    for word in line.split():
        if word not in d:
            d[word] = 1
```

```
else :
        d[word] += 1
l = list ()
for key, val in list (d.items()):
    l.append ((val, key))
~~d.sort(reverse~~
~~print(val~~
for val, key in l:
    print (val)
```

66. Explain advanced text parsing using dictionary (07 marks)

Answer The file can be parsed for words ~~with~~ in lower case & upper case as well as for punctuation. In order to do this two methods are used

1) lower() :- converts all upper case letter to lower case letter

2) Translate() :- Translate() method can be used to replace or delete a letter in a string. Translate() method takes maketrans() method as an argument.

maketrans() method takes 3 arguments. The third argument deletes the information in the string or a file.

example program

```
import string
fname = input (" Enter the filename : ")
try :
    fhand = open (fname)
except :
    print (" file cannot be opened : ", fname)
    exit()
d = dict ()
for line in fhand :
    line = line . rstrip()
    line = line . translate (line. maketrans
                                (" ", " ", string.
                                punctuation))
    line = line . lower()
    words = line . split()
    for word in words :
        if word not in d :
            d [word] = 1
        else :
            d [word] = d [word] + 1
print (d)
```

The above program converts all words in each line in a file to lowercase this resolve the problem of same word both in upper & lowercase. string. punctuation is a 3rd argument in maketrans which deletes all the punctuation in a file like , ! @ # etc in order to do this string is.

6c. Why search & find all functions of regular expressions used? Explain with suitable examples. (03 marks)

Answer search(), findall() are the functions of regular expressions that are used to search or match the pattern in the given string.

findall() function returns a list containing all the matches.

Ex:-
```
import re
txt = "The rain in Spain"
x = re.findall("ai", txt)
print(x)
```

```
["ai", "ai"]
```

Search() function searches string for a match, if there is a match it returns a match object

Ex:-
```
import re
txt = "The rain in Spain"
x = re.search("rain", txt)
print(x)
```

o/p
```
<re.match object: span=(4, 8),
match="rain">
```

Module - 4

7a. Define class. Explain classes and objects of Python in detail with suitable examples (10 Marks)

Answer: **Class** :- Class is a user-defined ~~fun~~ or programmer defined data type which set of variable and methods.

Class is a blueprint for creating objects. Class is an object constructor. Class is a prototype i.e class is like a factory for creating object. i.e any number of objects can be created.

Object: object is an instance of a class. A class is like a blueprint while an instance of class (object) is the copy of the class with actual values. An object consist of i) state ii) Behavior iii) Identity.

Syntax for creating class

```
class Classname:
    <Statement_1>
    ...
    ...
    <Statement_2>
```

Syntax for object instantiation

```
objectname = classname()
```

Attributes: Attributes are variables

There are two types of attributes

1) class attributes
2) object attributes

1) **Class attributes** :- are the attributes/variables that are defined directly in the class & those are shared by all objects of the class

Syntax to assign the value to class attribute

```
classname.classattribute = value
```

Syntax to access class attribute or variable is

```
objectname.classattribute
```

2) **Instance or object attribute** :- Attributes defined inside object are specific to object

Syntax to assign value to instance attribute

```
objectname.attributename = value
```

Syntax to access instance attribute

```
objectname.attributename
```

Example : create a class point an instantiate an object p also access the class attribute

```
class point:          # class declaration
    x = 5             # class attribute

p = point()           # object instantiation
print(p.x)            # accessing class attribute
```

o/p: 5

7b. What is pure function? Explain with an example — 06 marks

Ans: Pure function: 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.

Example: Create a class Time & instatite two time objects t_1 and t_2 and add two time objects.

```
Class Time:
    "Represents the time of a day"

def print_time(time):
    print("%.2d:%.2d:%.2d" % (time.hour, time.min,
                              time.second))

def add_time(t1, t2):
    sum = Time()
    sum.hour = t1.hour + t2.hour
    sum.min = t1.min + t2.min
    sum.second = t1.second + t2.second

    if sum.second >= 60:
        sum.second -= 60
        sum.min += 1

    if sum.second >= 60:
        sum.min -= 60
        sum.hour += 1

    return sum
```

```
t1 = Time()
t1.hour = 10
t1.min = 34
t1.sec = 25
print ("Time 1 is : ")
print_time (t1)

t2 = Time()
t2.hour = 2
t2.min = 12
t2.second = 41
print ("Time 2 is : ")
print_time (t2)

t3 = add_time (t1, t2)
print (" Sum of two time : ")
print_time (t3)
```

In this program t_1 attributes and t_2 attributes are not altered or modified by the user defined functions hence they are called as pure functions.

O/P : Time 1 is : 10 : 34 : 25

Time 2 is : 02 : 12 : 41

Sum of two time : 12 : 47 : 06.

8a. write a program with a function print_time that takes a time object and prints it in the form of hour: minute: second. Write another function is_after that takes two time object t_1 and t_2 and resume True if t_1 follows t_2 chronologically and False other wise. — 08 marks.

Ans:- Program:

```
class Time:
    " Represents time of the day"

def print_time (time):
    print ("%.2d: %.2d: %.2d" % (time.hour,
                                 time.min, time.second))

def is_after (t1, t2):
    if ((t1.hour, t1.min, t1.second) > (t2.hour,
                                        t2.min, t2.second)):

        print (True)

    else:
        print (False)

t1 = Time()
t1.hour = 11
t1.min = 59
t1.second = 30
print (" Time 1 is: ")
print_time (t1)
```

```
t2 = Time()
t2.hour = 02
t2.min = 15
t2.second = 30
print("Time2 is: ")
print_time(t2)
is_after(t1, t2)
```

O/P: Time1 is : 11:59:30

Time2 is : 02:15:30

True.

86. Write a note on operator Overloading with an example — (08 marks)

Answer: For every operator in python there is a corresponding special method.

for example: + operator in python corresponds to special method like __add__

as why to print something like string, python invokes method __str__

changing the behavior of an operator so that it works with programmer defined types is called as Operator Overloading.

Example program

```
Class Time:
    def __init__(self, hour=0, min=0, second=0):
        self.hour = 0
        self.min = 0
        self.second = 0

    def __str__(self):
        return ('%.2d:%.2d:%.2d' %(self.hour,
                                   self.min, self.second))

    def __add__(t1, t2):
        sum.hour = t1.hour + t2.hour
        sum.min = t1.min + t2.min
        sum.second = t1.second + t2.second

        if sum.second >= 60:
            sum.second -= 60
            sum.min += 1

        if sum.min >= 60:
            sum.min -= 60
            sum.hour += 1

        return sum

t1 = Time(9,35)
t2 = Time(1,45)

print(t1 + t2)
```

O/P: 11:20:00

Module - 5

9a. How to retrieve web pages using urllib? Explain how to compute frequency of each word in the file retrieved — 08 marks

Answer: In order to manually send & receive data over HTTP using the socket library, there is a much simpler way to perform this common task in python by using the urllib library.

urllib treat web page like a file. urllib handles all of the HTTP protocol & header details in order to retrieve web pages.

example :- To read file from web using urllib

```
import urllib.request
fhand = urllib.request.urlopen('filename.txt')
for line in fhand:
    print(line.decode().strip())
```

Once the webpage has been opened with urllib.urlopen, then it is treated like a file and read it through it using a for loop.

Program to retrieve the data from file & compute the frequency of each word in the file as follows

```
import urllib.request, urllib.parse, urllib.error
fhand = urllib.request.urlopen('filename.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. What is an API? Explain with a neat sketch — 08 marks

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

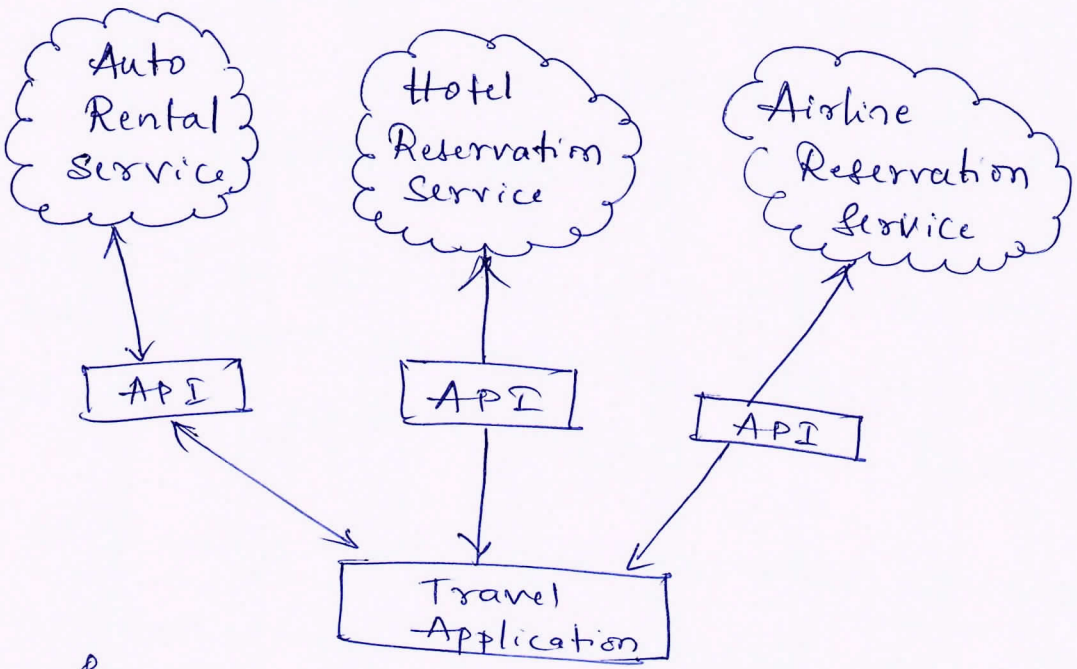

fig: Service Oriented Architecture

10a. Write a program to read binary file (08 Mark

Answer:

```
import urllib.request, urllib.parse, urllib.error
~~img = urllib.request.urlopen('filename.org')~~
img = urllib.request.urlopen('http://data.pr4e.org/
                                cover3.jpg').read()
fhand = open('cover3.jpg', 'wb')
fhand.write(img)
fhand.close()
```

This program reads all of data in at once across the network and stores it in the variable img in the main memory of computer, then opens the file cover.jpg and write the data out to your disk. This will work if the size of the file is less than the size of the memory of your computer

Suppose if this is a large audio or video file, this program may crash or at least run extremely slowly when your computer runs out of memory. We retrieve the data in blocks & then write each block to your disk before retrieving the next block.

The general name for these application-to-application contracts is Application Program Interfaces or APIs. When we use an API, generally one program makes a set of services available for use by other applications & publishes the API

When we begin to build our programs where the functionality of our program includes access to services provided by other programs, we call the approach a <u>Service-Oriented Architecture or SOA</u>.

A 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 standalone application which contains all the code necessary to implement the application.

Service-Oriented Architecture has many advantages including

- We always maintain only one copy of data
- Owners of the data can set the rules about the use of their data

With these advantage's an SOA system must be carefully designed to have good performance and meet the user's needs.

```
import urllib.request, urllib.parse, urllib.error
img = urllib.request.urlopen('http://data.pr4e.org/
                                cover3.jpg')
fhand = open ('cover3.jpg', 'wb')
size = 0
while True:
        info = img.read(100000)
        if len(info) < 1 : break
        size = size + len(info)
        fhand.write (info)
print (size, 'characters copied')
fhand.close()
```

10 b. Explain keys in a database model — (08 marks)

Answer: There are generally three kinds of keys used in a database model

1. Logical key: logical key is a key that the "real world" might use to look up a row. In our example data model, the name field is a logical key. It is the screen name for the user and we indeed look up a user's row several times in the program using the name field. You will often find that its make sense to add a UNIQUE constraint to a logical key.

2. <u>Primary key</u> :- It is usually a number that is assigned automatically by the database. It generally has no meaning outside the program and is only used to link rows from different tables together. When we want to look up a row in a table, usually searching for the row using primary key is the fastest way to find the row. Since primary keys are integers numbers, they take up very little storage & can be compared or sorted very quickly.

3. <u>foreign key</u> :- It is usually a number that points to the primary key of an associated row in a different table. An example of a foreign key in our data model is the from_id.