CBCS SCHEME

USN | | | | | | | | | |

BPLCK105B/BPLCKB105

# First Semester B.E./B.Tech. Degree Supplementary Examination, June/July 2024

# Introduction to Python Programming

Time: 3 hrs. Max. Marks: 100

*Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.*
*2. M : Marks , L: Bloom's level , C: Course outcomes.*

| | | Module – 1 | M | L | C |
|---|---|---|---|---|---|
| Q.1 | a. | Explain the following functions with examples:<br>i) input ii) print iii) len iv) str v) int | 10 | L2 | CO1 |
| | b. | Explain **if** and **elif** control statements with syntax and flowchart. | 5 | L2 | CO1 |
| | c. | Write a program to read name and year of birth of a person. Display whether the person is a senior citizen or not. | 5 | L3 | CO1 |
| | | **OR** | | | |
| Q.2 | a. | Explain the following with example:<br>i) Def Statements with Parameters<br>ii) Parameters and Return Values | 8 | L3 | CO1 |
| | b. | Explain the following, with syntax and example:<br>i) for loop ii) break iii) continue | 12 | L3 | CO1 |
| | | **Module – 2** | | | |
| Q.3 | a. | Define list. Explain append( ), index( ), sort( ) and insert( ) list methods with example. | 10 | L3 | CO2 |
| | b. | Read 10 numbers from a console and create a list. Develop a program to print the elements of created list, sorted list and reversed list. | 6 | L3 | CO2 |
| | c. | Explain copy( ) and deepcopy( ) functions of copy module. | 4 | L3 | CO2 |
| | | **OR** | | | |
| Q.4 | a. | Define dictionary. Explain the following methods of dictionary<br>i) setdefault ii) get iii) keys iv) items | 10 | L2 | CO2 |
| | b. | Write a program to count the number of occurrences of each letter in a given string. Use pretty print to format your output. | 10 | L3 | CO3 |
| | | **Module – 3** | | | |
| Q.5 | a. | Explain how individual elements of a string are accessed. How to extract a part of a string? Explain with examples. | 10 | L3 | CO3 |
| | b. | Explain any 5 string methods with syntax and example. | 10 | L3 | CO3 |

| | | | | | |
|---|---|---|---|---|---|
| | | **OR** | | | |
| **Q.6** | **a.** | Explain any 5 methods in os.path module related to files. | **10** | **L2** | **CO3** |
| | **b.** | Explain file reading and writing process with example. | **10** | **L3** | **CO3** |
| | | **Module – 4** | | | |
| **Q.7** | **a.** | Write a program to display folder name, list of subfolders, and files in the working directory using os.walk( ). | **5** | **L3** | **CO3** |
| | **b.** | Explain the following with respect to shutil module.<br>i) Copying files and folders<br>ii) Moving and renaming files and folders. | **8** | **L3** | **CO3** |
| | **c.** | Write a program to backup a folder into a ZIP file. | **7** | **L3** | **CO3** |
| | | **OR** | | | |
| **Q.8** | **a.** | What is an assertion? Explain how to use assert keyword with an example. | **7** | **L3** | **CO3** |
| | **b.** | Explain the different logging levels. | **7** | **L2** | **CO3** |
| | **c.** | Demonstrate reading and extracting from zip files using zipfile module. | **6** | **L3** | **CO3** |
| | | **Module – 5** | | | |
| **Q.9** | **a.** | Explain _ _ init_ _, _ _str_ _, _ _add _ _ methods with example. | **12** | **L3** | **CO4** |
| | **b.** | Explain type based dispatch with example. | **8** | **L3** | **CO4** |
| | | **OR** | | | |
| **Q.10** | **a.** | Define classes and objects. Write a program to create a class called student with attributes name, usn, sem, sec and create two student objects. Read and print the details of two students using appropriate methods. | **12** | **L3** | **CO4** |
| | **b.** | Explain pure functions with examples. | **8** | **L3** | **CO4** |

* * * * *

# Introduction to Python Programming

June/July 2024

Subcode: BPLCK105B/BPLCKB105 Time: 3 hrs

Max Marks: 100

## Module-1

Q1)

a) i) Input :- The input function waits for the user to type some text on the keyboard and press enter.

Ex:- myname = input()

ii) Print:- The print() function displays the String value inside the parentheses on the Screen.

Ex:- Print ("Hello world!")

O/P - Hello world!

iii) len :- The len() function evaluates to the integer value of the number of characters in that String.

Ex:- len ('Hello')

O/P → 5

iv) str:- The Str() function can be passed an integer value and will evaluate to a String value version of it.

Ex:- Str (29)

O/P → '29'

v) int:- The int() function converts the Specified value into an integer number.

Ex:- int(1.99)

O/P → 1

[illegible]

HOD
Computer Science & Engineering
K.L.S Vishwanathrao Deshpande
Institute of Technology, Haliyal

Q1)
b) Explain if and elif Control Statement with Syntax & flowchart.

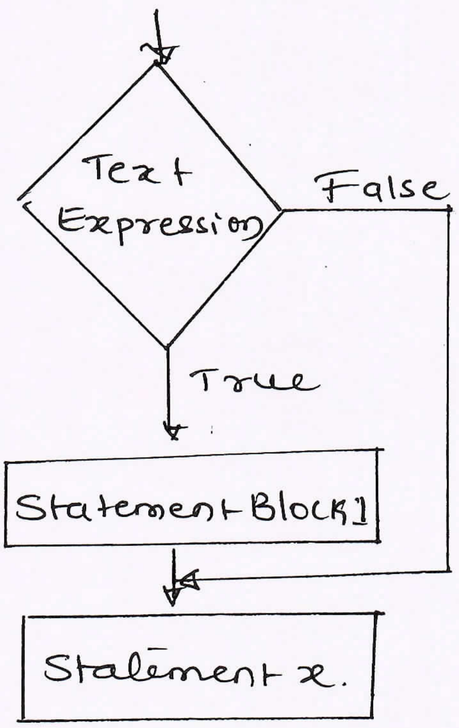
i) if Statement

if statement Consist of
*) The if keyword
*) Condition
*) Colon
*) Indentation.

Syntax

```
if text Expression:
     Statement 1
     - - - - - -
     - - - - - -
     Statement n
     Statement x.
```

Flow chart

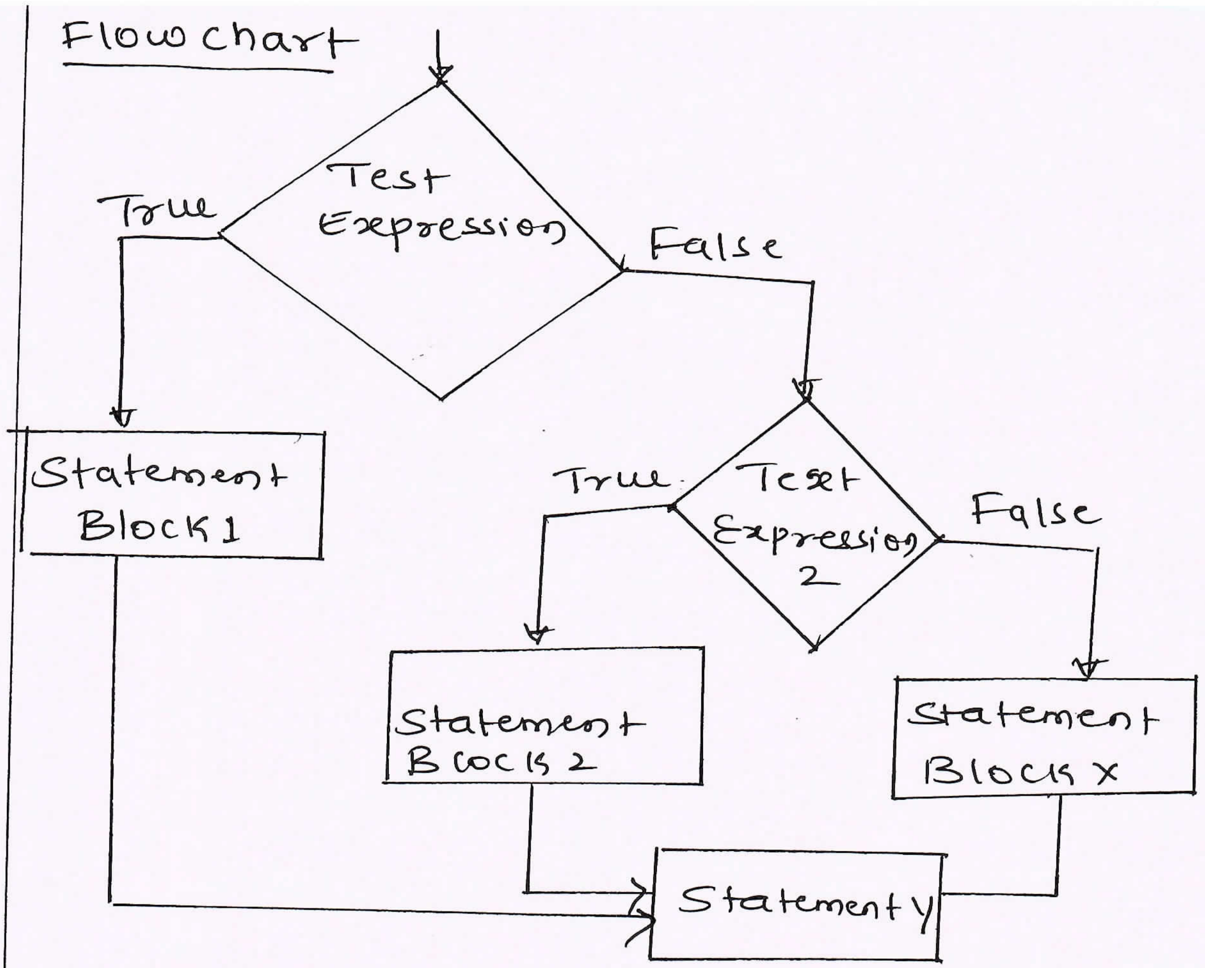


ii) elif Statement

Syntax

```
if (test Expression):
      Statement Block 1
elif (Text Expression):
      Statement Block 2
elif (Text Expression):
      Statement Block 3
      - - - - - - -
      - - - - - - -
else:
      Statement block x
Statement Y
```

Flow chart



Q1)
c) Write a Program to read name and year of birth of a person Display whether the Person is a Senior Citizen or not.

```
name = input()
~~age~~ = int(input("Enter your Year of Birth")
YOB
 age = 2025 - YOB
 if age > 60:
     Print ("you are Senior Citizen")
 else:
     print ("you are not Senior Citizen")
```

O/p

① 
```
name: xyz
YOB: 2000
age: 25
You are not Senior Citizen
```

② 
```
name: xyz
YOB: 1950
age: 75
You are Senior Citizen
```

Q2)
a) Explain the following with example:
i) Def Statements with parameters
ii) Parameters and Return values.

i) Def Statements with parameters

→ When we call the print() or len() function, we pass in values, called arguments in this context by typing them between the parentheses.

→ We can also define our own functions that accept arguments.

Ex:-

```
def hello (name):                       o/p
    Print ('Hello' + name)              Hello Alice
hello ('Alice')                         Hello Bob.
hello ('Bob')
```

→ The definition of the hello() function in this program has a parameter called name.

→ A parameter is a variable that an argument is stored in when a function is called.

→ One special thing about parameter is that the value stored in a parameter is forgotten when the function returns.

ii) Parameters and Return value

→ The value that a function call evaluates to is called the return value of the function.
Ex:- len('Hello') → Return value is 5

→ Return Statement consists of the following
1. The return keyword.
2. The value or Expression that the function should return.

Ex:-

```
def add_numbers (a, b):
        return a+b.
result=add_numbers (5,3)
Print (result)
```

o/p → 8.

Q2)
b) Explain the following, with Syntax & Example.
i) for loop ii) break iii) Continue.

i) for loop

Syntax

```
for loop_Control_Variable in Sequence
    Statement Block.
```

Ex:-

```
for i in range (1,6):
    Print(i)
```

o/p

```
1
2
3
4
5
```

ii) break

The break Statement is used to exit a loop Prematurely when a Specific Condition is met. It immediately terminates the loop & moves Control to the next Statement after the loop.

Syntax

```
for variable in sequence:
        if condition:
              break.

 while condition:
        if condition:
              break.
```

Ex:-

```
    numbers = [1, 2, 3, 4, 5]
    for num in numbers:
        if num==3:
            break
        print(num)
```

o/p →

```
1
2.
```

iii) Continue

The continue statement skips the current iteration of a loop and moves to the next iteration, without executing the remaining code in the loop body for that iteration.

Syntax

```
for variable in sequence:
       if condition:
            continue

while condition:
     if condition:
           continue.
```

Ex:-

```
 for num in range(1,6):
       if num ==3:
            continue
       print(num)
```

o/p →

```
1
2
4
5
```

Q3>
a> Define list Explain append() index() sort() and insert() list methods with example.

i> append() – The append method call adds the argument to the end of the list.

Ex:-
```
Spam = ['cat', 'rat', 'bat']
Spam. append ('dog')
Spam
```
o/p →
```
['cat', 'rat', 'bat', 'dog']
```

ii> index() → List value have an index() method that can be passed a value, and if that value exists in the list, the index of the value is returned if the value is not in the list, then python produces a Value Error.

Ex:-
```
Spam =['hello', 'hi', 'how']
Spam.index ('hello')
```
o/p → 5.
```
spam.index ('xyz')
```
o/p → Error.

iii> Sort() → List of number value or lists of strings can be sorted with the Sort() method

Ex:-
```
num = [2.5, 3.14, 1, -7]
num. Sort ()
num
```
o/p →
```
[-7, 1, 2.5, 3.14]
```

iv) insert() → The insert() method can insert a value at any index in the list.

Ex:-
```
Spam = ['cat', 'dog', 'bat']
Spam. insert (1, 'rat')
Spam
```
o/p →
```
['cat', 'rat', 'dog', 'bat']
```

Q3) b) Read 10 numbers from a Console and create a list. Develop a program to print the elements of Created list. Sorted list & reversed list.

```
number = [ ]
for i in range(~~0~~5):
numbers.append(int(input(" Enter number {i+1}")))
Print("Original list:-", numbers)
Print (" Sorted list:", Sorted (numbers))
Print ("Reversed list:", numbers [ : : -1])
```

O/P

```
Enter number 1:5
Enter number 2:2
   "        "  3:4
   "        "  4:0
   "        "  5:1.

Original list : [5,2,4,0,1]
Sorted list : [0,1,2,4,5]
Reversed list : [1,0,4,2,5]
```

Q3) c) Explain Copy() and deepcopy() functions of Copy module

Copy() function & Deepcopy() function

* If the function modifies the list or dictionary that passed we may not want these changes in the original list or dictionary value.
* for this, Python provides a module named Copy that provides both the Copy & deepcopy functions
* Copy.copy() Can be used to make duplicate Copy to mutable value like a list or dictionary.

Example for Copy() function

```
import Copy
Spam =['A','B','C','D']
Cheese = Copy.Copy (Spam)
cheese[1] =42
Print (spam)
Print (Cheese)
```

Example for deep Copy() function

```
import Copy
old_list =[[1,1,1],[2,2,2],[3,3,3]]
new_list = Copy.deepCopy (old_list)
old_list[1][0] = 'BB'
Print ('old_list': old_list)
Print ('new_list': new_list)
```

Q4)

a) Define dictionary. Explain the following methods of dictionary.
i) Setdefault ii) get iii) Keys iv) items.

Dictionary is a value that Contains multiple non- ordered Sequence.

(i) Set default() method

It will take 2 arguments
1. The Key is check for.
2. The value to set that Key if the Key doesnot Exist.

Ex:-
```
Spam = {'name': 'Raju', 'age': '5'}
if 'color' not in Spam:
        Spam['color'] = 'black'
        Print (Spam)
```

ii) get method()

It takes 2 arguments

* The key of the value to retrieve
* Fall back value to retrieve if that key does not Exist.

Ex:- picnic items = {'apple':'5', 'cup':'2'}.

```
'I am bringing'+str(picnicitems.get('cup',0)+'cups
```

iii) keys() - The keys() method returns a view object the view object contains the keys of the dictionary or a list.

Ex:-

```
Spam ={'color':'red', 'age':'42'}.
for k in Spam.keys():
        Print(k)
```

(iv) items() - The items() method in Python returns a view object that contains a dictionary key-value pairs or list of a tuples.

Ex:-

```
Spam ={'color':'red', 'age':'42'}.
for i in Spam.items():
        Print(i)
```

Q 4 b) Write a program to Count the no of occurance of each letter in a given String use prety print to format your output. ⑥

```
import pprint
message = 'Hello how are you'
Count = { }
for character in message:
        Count.Setdefault (character, 0)
        Count[character] = Count [character] +1
Pprint.Pprint(count)
```

<u>o/p</u>

```
' ' : 3
'a' : 1
'e' : 2
'H' : 1
'h' : 1
'l' : 2
'o' : 3
'r' : 1
'u' : 1
'w' : 1
'y' : 1
```

Q5) a) Explain how individual elements of a String are accessed. How to extract a part of a string? Explain with example.

A String in python is a Sequence of characters, and individual elements can be accessed using indexing

Indexing in Strings

* Python uses zero-based indexing, meaning the first character has an index of 0, the Second 1 and so on.
* Negative indexing starts from -1 (last character) -2 (Second last) etc.

Ex:-
```
text = "python"
    Print(text[0])
o/p → p.
    Print(text[-1])
o/p → n
```

Extracting a part of a string

Slicing allows Extracting a portion of a String using the Syntax:

String[start: end: step]

Ex:-
```
text = "pythonprogramming"
    Print(text[0:6])
o/p → python.

    Print(text[-11:-7])
o/p → rgorp
```

Q5)
b) Explain any 5 String methods with Syntax & Examp (4)

(i) isalpha()

It returns the true if the String Consist of only letters & not having blank.

Ex:- 'Hello'.isalpha()
o/p → True.

(ii) upper() and lower()

String methods return a new string where all the letters in the Original String have been Converted to uppercase or lowercase, respectively.

```
Ex:- Spam = 'Hello world!'
   >> Spam = Spam. upper()
   >> Spam
   . 'HELLO WORLD!'

  >> Spam = Spam.lower()
  >> Spam
     'hello world!'
```

(iii) isupper() and islower

String method will return a Boolean True value if the String has at least one letter and all the letters are uppercase or lowercase respectively otherwise, the method returns False.

Ex:-
```
   >>> Spam.islower()
  o/p → False
   >>> 'HELLO'.isupper()
  o/p → True.
```

(iv) isalnum() – Returns True if the String Consists only of letters & numbers & is not blank.

Ex:- 'hello123'.isalnum()
o/p → True.

(V) isdecimal – Returns True if the String Consists only of numeric characters and is not blank.

Ex:- '123'.isdecimal()
o/p → True.

Q6>
a> Explain any 5 methods in os.path module related to files.

(i) os.path.abspath()
Will return a String of the absolute path of the argument
Ex:- os.path.abspath(' ')
o/p → 'c:\\ Python 34'

(ii) os.path.isabs(path)
Will return true if the argument is an absolute path & false if it is a relative path.
Ex:- os.path.isabs('.')
o/p – False

(iii) os.path.relpath (Path, Start)
Will return a String of relative path from the Start path to path.
if Start is not provided the Current Working directory is used as the start path.
Ex:- os.path.relpath ('c:\\windows', 'c:\\')
o/p → windows.

(iv) os.path.dirname(path)
Will return a string of Everything Comes before the Slash in the path argument
Ex:- Path = 'c:\\window\\System 32\\ Calc.Exe'
os.path.dirname (path)
o/p
'c:\\window\\ System32'

(v) os.path.basename(path)

Will return a String of everything that Comes after the last Slash in the path argument

Ex:- os.path.basenames(Path)

o/p → 'calc.Exe'

Q6) b) Explain file reading and writing process with Example.

1) Open file with open() function

TO open a file with the open() function, You pass it a String path indicating the file you want to open: it can be either an absolute or relative path.

Ex:-

```
helloFile = open('C:\\Users\\your_home_folder\\
                     hello.txt')
```

→ Read mode is the default mode for files you open in python.

→ when a file is opened in read mode, python lets you only read data from the file: You can't write or modify it in any way.

2) Writing File

→ Python allows you to write content to a file in a way Similar to how the print() function 'writes' Strings to the Screen.

→ Write mode will overwrite the Existing file and Start from Scratch, just like when you overwrite a Variable's Value with a new Value.

→ pass 'w' as the Second argument to open() to open the file in Write mode.

Ex:- baconFile = open('bacon.txt', 'w')

Q 7)
a) Write a program to display folder name, list of Subfolders and files in the working directory using os.walk()

```
import os
for foldername, subfolder, filename in os.walk('c:\\delicio
        print('The current folder is'+folder name)
for filename in filename:
        print('File insider'+folderName+', '+filename)
        print('.')
```

O/P

```
The current folder is C:\\delicious
Subfolder of C:\\delicious : cats
Subfolder c:\\delicious : spam.txt.
The current folder is c:\\delicious\\cats
file inside c:\\delicious\\cats : Catnames.txt.
fileinside c:\\delicious\\cats : Zophie.jpg.
```

Q7)
b) Explain the following with respect to Shutil module
i) Copying files & folders
ii) Moving & renaming files & folders

i) Copying files & folders.

* Copying files as well as entire folders by calling function shutil.copy (source, destination)
* It will copy the file at path source to the folder at the path destination.
* If a destination is a filename that filename will be used as the new name of the copied file

Source → C:\\ Spam.text

destination → C:\\delicious

o/p → 'C:\\delicious\\ Spam.txt.

Ex:-

```
import shutil.os
os.chdir('C:\\')
shutil.copy('C:\\Spam.txt', 'C:\\delicious')
```

o/p → 'C:\\delicious\\ Spam.txt'

shutil.copytree() will copy an entire folder & every file contained in it.

Ex:-

```
import shutil.os
os-chdir('C:\\')
shutil.copytree('C:\\ bacon', 'C:\\ bacon_
                                   backup')
```

o/p → 'C:\\bacon_backup'

ii) Moving & Renaming files & folders

* Calling shutil.move will move the file or a folder at the path source to the path destination & will return & String of the absolute path of the new location.

* If destination points to a folder the source file gets moved into a destination & keep it current file name.

Ex:-

```
import shutil
shutil.move('C:\\ bacon.txt', 'C:\\apple')
```

o/p → 'C:\\apple\\ bacon.txt'

Q7)
c) Write a program to backup a folder into a ZIP file.

```
from zipfile import ZipFile
import os. zipfile
extension.input (Input file Extension)
Zippy=Zipfile ('Backup.zip', 'w')
for folder Subfolders file in os.walls ('C:\\
            ADMIN-3-24-48\\ untitled folder'):
    for Subfolder in Subfolders:
        path = folder + subfolder.
    for x in file:
        if x. endswith (Extension):
            filepath. folder + "\\" + x
            Print (filepath)
            Zippy.write (filepath, compress_type
            Zipfile, ZIP_DEFLATED)
Zippy.Close()
```

O/P → Backup.zipfile is Created.

(Q8)
a) What is an assertion? Explain how to use assert keyword with an example. (10)

* An assertion is a sanity check to make sure your code is not doing something obviously wrong.
* Sanity checks are performed by assert statement if the sanity check fails then assertion Error exception raise.
* It consists of
1) The assert keyword.
2) Condition.
3) A string to display when the condition is False.

Syntax

assert Condition, optional_message

Ex:-

```
  def divide(a,b):
      assert b!=0, "Denominator Can not be Zero"
      return a/b
 Print(divide(10,2))   # works fine
 Print(divide(5,0))    # Raises Assertion Error
```

O/P

```
5.0
Traceback (most recent Call last):
  File "Example.py", line 6
 assert b!=0, "Denominator Can not be Zero"
 Assertion Error: Denominator Can not be Zero.
```

Q8)
b) Explain the different logging levels.

| Level | Logging function | Description |
|---|---|---|
| DEBUG | logging.debug() | *Lowest level used for Small Details *Usually you Care about these messager only when dignosing Problems |
| INFO | logging.info() | *Used to record information on geners events in your Program or Confirm that things are working at their Point in the program |
| WARNING | logging.Warning() | *Used to indicate a Potential problem that does not Prevent the Program from Working but might doso in the future |
| ERROR | logging.Error() | *Used to record an Error that Count the Program to fall to do Something. |
| CRITICAL | logging.Critical() | *The highest level used to indicate a folder Error that has Caused or is about to Course the program |

Q8) C) Demonstrate reading and Extracting from zipfiles using zipfile method.

1) Reading zipfiles

→ To read contents of a zipfile, first you must create a zipFile object.

→ ZipFile objects are conceptually similar to the File objects you saw returned by the open() function

Ex:-
```
import zipfile,os
os.chdir('C:\\')
exampleZip = zipfile.ZipFile('Example.zip')
exampleZip.namelist()
['spam.txt', 'cats/', 'cats/zophie.jpg']
spamInfo = exampleZip.getinfo('spam.txt')
~~spamInfo.compress_size.~~
spamInfo.file-size
13908
spamInfo.compress_size
3828.
```

2) Extracting from ZIP Files

→ The Extractall() method for ZipFile objects extracts all the files & folders from a ZIP file into the current working directory

Ex:-
```
import zipfile,os.
os.chdir('C:\\')
examplezip= zipfile.ZIPFile('example.zip')
exampleZip.extractall()
exampleZip.close()
```

Q9)
a) Explain __int__, __str__, __add__ methods with example.

1) Initialization method (__int__ method)

The initialization method is a Special metho involved when the object is Created. The name of this method is __int__.

Ex:-
```
Class Time:
    def__int__(Self, hours=0, minutes=0, Seconds=0)
        Self.hours = hours
        Self.minutes = minutes
        Self. Seconds = Seconds
Current Time= Time(9,14,30)
Current Time = printTime()
```
O/P → 9:14:30.

>> the argument are Optional we can omit them.
```
CurrentTime = Time()
CurrentTime. PrintTime()
```
O/P → 0:0:0

2) __str__:

The next method __str__ returns a String representation of a point object, if a class provider, a method named __str__ it overrides the default behaviour of the Python built in str function.

Ex:-
```
Class Car:
    def __int__(Self, brand, model)
        return ('{Self, brand}, {self, model}')
        my_Car = Car ("Toyota", "Innova")
Print (my_car)
```
O/P → Toyota Innova

3> __add__:

For example to override the addition Operator (+) we provide a method __add__

Ex:- 
```
class point:
    def __add__ (self, other):
        return Point(Self.x + other.x,y) Self.y
                                         + other.y)
        P1 = point (3, 4)
        P2 = Point (5, 7)
        P3 = P1 + P2
    Print P3
```
O/p → (3+5, ~~4~~ + 7)

(8, 11)

Q9>
b> Explain type based dispatch?

→ Type based dispatch is a technic where a function or a method behave differently based on the type of the input argument.

Ex:-
```
    Start = Time (9, 45)
    duration = Time (1, 35)
>> Print (Start + duration)
```
O/p
```
    11:20
```
```
>> Print(Start + 1337)
```
O/p
```
    10:7:28.
```

Q10)
a)

```
Class Student:
        Marks = [ ]
        def getdata(Self, rn, name, m1, m2, m3):
                Student, rn = rn
                Student, name = name
                Student, Marks . append (m1)
                Student, marks . append (m2)
                Student, marks . append (m3)
      def total (Self):
                return [Student . marks[0] + Student.
        Marks[1] + Student . marks[2] + Student.Marks[3]
      def average(Self):
            Print ('Roll number is :', Student, rn)
            Print('Name is :', Student, name)
            Print ('marks are :', Student . marks)
            Print ('Total marks are :', self.total())
            Print ('Average marks are :', Self.average())
Print('Enter the roll number')
r = int (input())
name = input ("Enter the name :')
m1 = int (input ('Enter the marks in the first Subject'))
m2 = int(input('Enter the marks in the Second Subject))
m3 = int (input('Enter the marks in the third Subject'))
S1 = Student()
S2 = getData(rn, name, m1, m2, m3)
S1 . display Data ( )
```

O|P

```
Enter the roll number
201
Enter the name: xyz
Enter the marks in the first Subject: 98
Enter the marks in the Second Subject: 95
Enter the marks in the third Subject: 99
Roll number is : 203
Name is: xyz
marks are [98, 55, 99]
Total Marks are : 292
Average marks are : 97.33
```

Q10) b) Explain pure function with example?

→ The function Creates a new time object, initializes its attributes & returns a reference to the new object, this is Called a pure function because it does not modify any of the objects Passed to it an arguments & it has no sides effects, Such as displaying a value or getting user input.

Ex:-

```
def addTime(t1, t2):
        Sum = Time()
        Sum.hours = t1.hours + t2.hours
        Sum.minutes = t1.minutes + t2.minutes
        Sum.Seconds = t1.Seconds + t2.Seconds
return Sum
```

[illegible]

Prof. Santosh Savanur

[illegible]

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