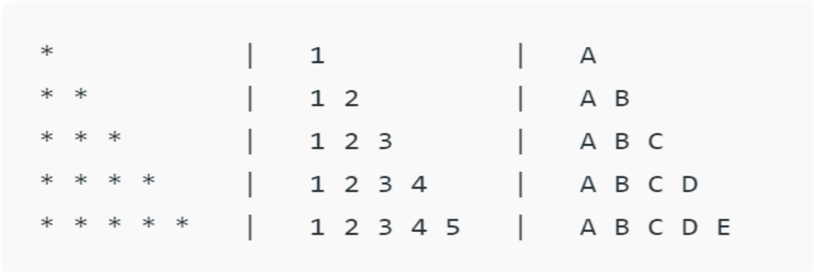
Laboratory Manual

(Python Lab BCA29106)

Table of Contents

<table>
<tr>
<td></td>
<td>to n.
IX. Write a python program to print the following pattern-

<pre>
* | 1 | A
* * | 1 2 | A B
* * * | 1 2 3 | A B C
* * * * | 1 2 3 4 | A B C D
* * * * * | 1 2 3 4 5 | A B C D E
</pre>
X. Write a program Using a while loop to find and print the sum of all prime numbers less than or equal to n.
XI. Write a program to Implement a while loop to repeatedly ask the user for a number until they enter a negative number. For each entered positive number, calculate and print its factorial.</td>
<td></td>
</tr>
<tr>
<td><b>2. Lists</b></td>
<td>I. Write a program in Python to create a list and print the value of the list.
II. Write a program in Python to create a list of integers then separate odd and even numbers from the given list.
III. Write a program in Python to search for an item from the list.
IV. Create a program in Python to create a list then add some items to the list and delete items from the list.
V. Write a Python program to perform matrix(3x3)</td>
<td>3</td>
</tr>
</table>

	multiplication using a list. VI. Write a Python program to demonstrate the different functions of a list. VII. Write a program in Python to demonstrate tuples. VIII. Write a program in Python to demonstrate string and its functions. IX. Write a python program to print all the characters(A-Z). X. Write a python program to check whether a substring is present or not in a string.	
3. Function & Modules	I. Write a PYTHON program to check whether a number is prime or not using function. II. Write a PYTHON program to check if a number is a palindrome or not using function. III. Write a PYTHON program to calculate the Fibonacci series using function. IV. Write a PYTHON program to swap two numbers by call by value and call by address. V. Write a program to find LCM and GCD of two numbers using user define function. VI. Write a Python program to create a dictionary and update existing key values. VII. Write a Python program to sort (ascending and descending) a dictionary by value. VIII. Write a Python program to concatenate the following dictionaries to create a new one. IX. Write a Python program to check whether a given key already exists in a dictionary. X. Write a Python program to change the value of a specific item by referring to its key name. XI. Write a Python program to check all values are the same in a dictionary. XII. Python Program to Count the Frequency of Words Appearing in a String Using a Dictionary	3

4. File handling , oops and event driven programming.	I. Write a program to create a text file named 'std_rec' with record stu_name,stu_roll and stu_marks. II. Write a python program in python to create a binary file and print the content of the file. III. Write a program to demonstrate Object and Class in Python. IV. Write a program to demonstrate Inheritance in python. V. Create a GUI program for calculator using python tkinter. VI. Create a GUI program for Calendar using python tkinter.	3-4
5. Packages	I. Write a program to element-wise addition of two numpy arrays. II. Write a program for multiplying two matrix(2D numpy array). III. Create a program to create a line graph for two different array using matplotlib. IV. Write a program in matplotlib for creating a Pie chart. V. Program to demonstrate Scipy and its functions.	**4**

Aim/Purpose of the Assignments:

Assignment1: Aim of this assignment is to learn about the basic concept of programming language and gathers the basic knowledge of Python programming language. Also learn about the variables and how to implement various operators. decision making in Python is to selectively execute code based on specified conditions, facilitating dynamic and adaptive program behavior. We also know about the loops. loops are responsible for performing repetitive tasks using a short code block that executes until the condition holds true.

Assignment2: The aim of lists in Python is to provide a versatile and dynamic data structure that allows the storage and manipulation of ordered collections of items, facilitating tasks such as iteration, modification, and retrieval of elements within a program. Lists enable the representation of sequences, making them fundamental for various data processing and manipulation tasks.2-D arrays in Python is to extend this functionality to two dimensions, organizing elements in rows and columns. This allows for the representation of tabular data and matrices, making it suitable for tasks such as numerical computations.

Assignment3: The aim of understanding dictionaries in Python is to efficiently store and retrieve key-value pairs, providing a flexible and dynamic data structure for various programming tasks. functions and modules in Python is to modularize code for reusability, maintainability, and abstraction, while promoting efficient and organized development practices. using functions with parameters in Python is to create reusable and flexible code by allowing the passing of input values, enabling customization and versatility in function behavior.

Assignment4: The aim of understanding file handling for input/output operations, object-oriented programming concepts like class, object, and inheritance, as well as event-driven and GUI programming in Python, is to develop versatile applications that efficiently manage data, implement modular and organized code structures, and create interactive graphical user interfaces for enhanced user experiences.

Assignment5: The aim of packages like NumPy, Pandas, and Matplotlib in Python is to provide powerful and specialized tools for efficient numerical computing, data manipulation, and visualization, enhancing productivity and capabilities in scientific and data-driven applications.

1. Learning Outcomes:

After the completion of the course, students would be able to:

CO1: Define knowledge about the basic concept of writing a program and summarize the role of constants, variables, identifiers, operators, type conversion and loops, decision making Python Language.

CO2: Understand and illustrate the list and matrix representation in python.

CO3: Analyze the role of Functions involving the idea of modularity and construct user defined functions.

CO4: Illustrate error handling techniques using exception handling and develop python programs using file input/output operations.

CO5: Evaluate and criticize the different package with different types of programmes in python.

2. Prerequisites:

Basic functions of computer and basic knowledge of programming.

3. Software required:

Python IDE/VS code/Online python compiler.

4. Introduction and Theory:

Assignment 1: Introduction to python programming

Python is a dynamic typed language.Variables in python is define as –

```
x=10
y='Raja'
z=5.6
```

where x, y and z are integer, string and float-type variables respectively.

Conditional Statements-

Syntax of if else: : The if-else statement in Python is a flow control statement used for decision-making in the Python program

```
    If(condition):
        #Body of the loop
    else:
        #Body of else
```

Syntax of if..elif..else : The if-else statement in C is a flow control statement used for decision-making in the C program.

```
If(condition-1):
    #body of if
elif(condition-2):
    #body of elif
else:
    #body of else
```

Syntax of for loop: loop is used to repeat a set of instructions based on a condition.In python we use for and while loop.

```
    For I in range():
        #body of the loop
```

Syntax of while loop:

```
    Initialization
    While(condition):
        #body of the loop
        updation
```

Assignment 2: List

List is datatype in python which can store various items of different data types.

Syntax of List:

```
    Name_of_list=[‘item-1’,’item-2’,……’item-n’]
```

It create the list with the given elements.

The list can be traverse as follows:

```
        For I in Name_of_list:
            Print(i)
```

Nested list can be define as follows:

```
        Nested_List=[[10,20,30]
                    [30,40,50]]
```

Functions related to list:

```
   List(),len(),find(),append(),insert(),ex
tent(),sort(),sorted(),copy(),pop(),remove(
),reverse() etc
```

Tuple:

Tuple is a immutable sequence data type in python.It also can store multiple items of different type.

Syntax of tuple:

```
            Name_of_tup=("item_1","item-
2"……"item-n")
```

For traversing the tuple-

```
            For I in Name_of_tup:
                Print(i)
```

String:

String is a sequence datatype.Strings are enclosed between single or double quotes.

Syntax-

```
            Var_name="Raja"
```

Traversal of strings:

```
            For I in Var_name:
                Print(i)
```

String functions:

`Len(),find(),index(),replace(),lower(),upper(),isalpha(),isnum(),title(),islower(),isupper(),isdigit(),endswith(),startswith(),split(),etc`

Assignment 3:Functions and Module

Functions are two type one is build-in function and other is user-define function.

Syntax for defining user define function:

```
        def fun_name(list_of parameters):
            #body of the function
            #return statement
        fun_name(argument_list) #function call
```

here def is the keyword used to create a user define function.and fun_name is the name of the function.

Modules in python:Collection of related function is called module.In python there are many build in modules.For use these module in program we use import statement as follows.

```
Import math *
Or,from math import *
```

Where math is a module.

Assignment 4:File Handling and GUI:

In python we use mainly three types of files i.e, text file, binary file and csv file.

Syntax for creating a text file:

```
   Fp=open(“name_of_file”,”open_mode”)
          fp.write(“contents_of_the_file”)
#write to the file
          read_data=fp.read(n)
#read_from_the_file
          print(read_data)
#print_the_dead_data
```

Tkinter:Tkinter is a python build-in module used to create python GUI applications.

Basic Syntax of Tkinter:

```
import tkinter
      m = tkinter.Tk()
      #widgets are added here
```

```
m.mainloop()
```

Object Oriented Programming(OOP's)-

Syntax for creating object and class:

```
class Person:
    def __init__(self, name, age):
    self.name = name
       self.age = age

p1 = Person("John", 36)#p1 is an
object of class Person.

print(p1.name)
print(p1.age)
```

Assignment 5:

Numpy Library: NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Syntax of Numpy array:

```
import numpy as np
 a = np.array([2,3,4])
```

```
print(a)
```

Matplotlib Library: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

```
from numpy import *
 from pylab import *
 x = linspace(-3, 3, 30)
 y = x**2
plot(x, y)
show()
```

Scipy Library: SciPy is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

Syntax of using scipy:

```
    import scipy
    arr1 = scipy.mean([1, 3, 27])
    print("Arithmetic Mean is :",
arr1)
```

5. **Operating Procedure:**

Step 1: Create the Python File

Open a Python compiler or Code block on your computer.

Write the Python code for your array and Linear and binary searching in the text editor.

Save the file with a ".py" extension. Choose a meaningful filename for your corresponding codes.

Step 2: Run the saved File

Locate the saved file on your computer.

Precautions and/or Troubleshooting

1. Code Review: Check your code carefully for logical problems, syntactic faults, and any vulnerabilities before executing your program.

2. Compiler Warnings: Be mindful of mistakes and warnings from the compiler. By taking care of these problems, you can keep your application free from bugs and unexpected behavior.

3. Use Version Control: Use a version control system (e.g., Git) to track changes in your code. This helps you revert to a working version if something goes wrong.

4. Input Validation: Always validate user inputs and ensure they are within expected ranges. This prevents crashes and security vulnerabilities like buffer overflows.

5. Boundary Checks: Avoid accessing arrays or pointers beyond their boundaries, as this can lead to memory corruption and crashes.

6. Use Libraries: Whenever possible, use well-tested and widely used libraries to handle common tasks. This reduces the chances of reinventing the wheel and introducing bugs.

7. Debugging Tools: Learn to use debugging tools such as gdb to locate and resolve problems. You can examine the behavior of the software step-by-step with the use of these tools.

8. Logging and Error Handling: Implement proper logging and error-handling mechanisms. This makes it easier to identify issues and their causes when your program is in production.

9. Code Modularization: Break your code into smaller, manageable modules. This promotes code reusability, ease of maintenance, and makes troubleshooting more manageable.

10. Test Cases: Create comprehensive test cases that cover various scenarios and edge cases. Automated tests help catch regressions when making changes to the code.

11. Testing Environments: Test your program in different environments (e.g., different operating systems, hardware configurations) to ensure portability and compatibility.

12. Plan for Scalability: If your program might need to handle larger datasets or higher loads in the future, consider designing it in a way that can easily scale without major modifications.

13. Security Considerations: Be aware of potential security vulnerabilities like race conditions. Sanitize inputs and use secure coding practices.

14. Documentation: Document your code, including its purpose, functions, inputs, outputs, and potential pitfalls. Well-documented code is easier to troubleshoot and maintain.

15. Backup and Versioning: Maintain many versions of your code and regularly backup your work, especially before making big changes. This guarantees that you can return to a functional condition if needed.

6. Observations

Try using various methods to implement and observe the output

7. Calculations & Analysis

Check the output without error.

8. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

9. Extension and Follow-up Activities (if applicable)

Not Applicable

10. Follow Up Question:

a. Given a function that does not return any value, What value is thrown by default when executed in shell.

b. Following set of commands are executed in shell, what will be the output?

1. >>>str="hello"
2. >>>str[:2]
3. >>>

c. What error occurs when you execute? apple = mango

d. Carefully observe the code and give the answer.

1. def example(a):
2. a = a +'2'
3. a = a*2
4. return a
5. >>>example("hello")

e. Select all options that print hello-how-are-you

a) print(‘hello’, ‘how’, ‘are’, ‘you’)
b) print(‘hello’, ‘how’, ‘are’, ‘you’ + ‘-‘ * 4)
c) print(‘hello-‘ + ‘how-are-you’)
d) print(‘hello’ + ‘-‘ + ‘how’ + ‘-‘ + ‘are’ + ‘you’)

f. The value of the expressions 4/(3*(2-1)) and 4/3*(2-1) is the same. State whether true or false.
g. How can you pick a random item from a range?
h. What Are Decorators In Python?
i. How will you convert a string to all lowercase?
j. How will you get the min alphabetical character from the string?
k. What is the output of ['Hi!'] * 4?
l. How Do You Traverse Through A Dictionary Object In Python?
m. How Do You Delete Elements Of A Dictionary In Python?
n. What Is The Syntax For List Comprehension In Python?

o. Suppose listExample is [‘h’,’e’,’l’,’l’,’o’], what is len(listExample)?

p. How will you get the max valued item of a list?

q. Does python support multiple inheritance?

r. Name the python Library used for Machine learning.

s. What Are The Different Methods To Copy An Object In Python?

t. What is Keyword in Python?

u. What is the output of the below code?

```
 >> import array >>> a = [1, 2, 3]
>>> print a[-3]
>>> print a[-2]
 >>> print a[-1]
```

v. What is the output of the below program?

```
>>>names = [‘Chris’, ‘Jack’, ‘John’, ‘Daman’]
 >>>print(names[-1][-1])
```

w. What is lambda in Python?

x. What is namespace in Python?

y. What is module and package in Python?

z. Illustrate the proper use of Python error handling ?

aa. Name the four main types of namespaces in Python ?

bb. What is pickling and unpickling?

cc. What is the output of the following? print("xyyzxyzxzxyy".count('yy'))

dd. What is the output of the following? print("xyyzxyzxzxyy".count('xyy', -10, -1))

ee. What is the output of the following code?

```
 a={1:"A",2:"B",3:"C"}
a.clear()
print(a)
```

11. Assessments

As per the assessment and evaluation policy of university

12. Suggested readings

1. Python Programming,Oxford publication,Reema Thareja.
2. Python CookBook,O'reilly Publication, *David Beazley and Brian K. Jones,3E.*

13. Assignment copy format:

1. Write in A4 page; No channel file will be accepted.

2. Front page and Index page format will be provided by department
3. Following points must be included while writing assignment copy
    a) Problem definition
    b) Algorithm
    c) Program in corresponding language
    d) Output