Department of Computer Science and Engineering

Laboratory Manual

of

IT3008 - Programming with Python

is submitted to

Mr. Vishvajit Bakrola

Assistant Professor

Asha M. Tarsadia Institute of Computer Science and Technology

Uka Tarsadia University, Maliba Campus Bardoli, Gujarat

Semester – 2

(Summer 2022)

ASHA M. TARSADIA INSTITUTE OF
COMPUTER SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that Mr/Ms.____, Enrollment No: ___________________________ of B.Tech. Computer Science and Engineering 2nd semester has satisfactorily completed his/her laboratory work of **IT3008 - Programming with Python** during regular term in academic year 2021-22.

Date of Submission: _______________

Mr. Vishvajit Bakrola
Subject Teacher
Assistant Professor
Department of Computer Science and
Engineering

Department Authority

Institute Stamp

IT3008 - Programming with Python
Laboratory Practical Index

Sr. No.	Name of Practical	Actual Date
1	Write a python program to greet users with welcome messages using print() method.	21/02/2022
2	Write a python program to demonstrate the creation of List data structure along with its methods - append(), extend(), insert(), remove(), clear(), index(), count(), sort(), reverse(), and copy(). I. Demonstrate positive and negative indexing with python List. II. Demonstrate slicing operations on python List. III. Demonstrate updation on List elements in python. IV. Demonstrate deletion of a single python list element and multiple elements using slicing operator.	21/02/2022
3	Write a python program to demonstrate the creation of tuples along with its methods - count() and index(). I. Demonstrate positive and negative indexing with python Tuple. II. Demonstrate slicing operations on python Tuple.	28/02/2022
4	Write a python program to demonstrate the creation of a Dictionary *student* with the *name*, *age* and *branch* of a student. I. Demonstrate the updation of python dictionary. II. Demonstrate the removal of elements from the python dictionary. III. Demonstrate the use of following dictionary methods - clear(), copy(), get(), items(), keys(), popitem(), and values().	28/02/2022
5	Demonstrate the use of basic string methods in python - lower(), upper(), join(), split(), find(), and replace().	07/03/2022
6	Write a python program to implement basic arithmetic operations on user entered numbers.	07/03/2022
7	Write a python program to count how many times a specific number is occurring in a list. Take user input for both numbers and a list.	07/03/2022
8	Write a python program that takes N number of integers from the user in a	14/03/2022

	python list. Create a function that takes the list of user entered numbers and returns MAX and MIN numbers from that list to the user.	
9	Write a python program to perform basic matrix operations on user entered matrices.	14/03/2022
10	Write a python program to perform basic operations of a calculator. Provide choice for operations to users and make a program iterative. Provide specific exit option to users.	21/03/2022
11	Write a python program to demonstrate the use of arbitrary arguments.	21/03/2022
12	Create a class named *student* having attributes - *std_name*, *std_age*, *std_branch*, and *std_city*. Create a method named *get_data*() in *student* class that takes user input for these attributes and a method named *display*() that prints the attribute values on the terminal. Call both the methods by creating an instance *std_obj* of the class *student*.	21/03/2022
13	Write a python program to demonstrate basic banking operations. Create a class named banking having separate class methods for each operation. Call each method with an instance of the class and attribute values to be taken from the user.	28/03/2022
14	Create a class named *employee* having attributes - *emp_name*, *emp_age*, and *emp_city*. Create a method named *get_data*() in employee class that takes user input for these attributes. Derive a class named *emp_derived*() from the employee class, having an __*init*__() method that displays the attributes of the employee class upon instantiation.	28/03/2022
15	Create a base class named *university* with its attributes - *name, year_of_estd*, and *city*. Derive following class from the super class *university*: *professor*, *lab_assistant*, *office_assistant*, and *peon*. Make the program choice based on the user. The attributes and method of various class are as below: - Attributes of *professor* class: *designation*, *highest_qualification*, *area_of_research, year_of_joining, year_of_ experience*, and *name_of_institute*. - Methods of professor class: __*init*__() method that gets invoked upon instantiation and takes values of class attributes. The *display*() method that prints class attribute values along with attributes of its super class. - Attributes of *lab_assistant* class: *designation* = “Lab Assistant” (static), *highest_qualification*, *additiobnal_skilss*, *year_of_joining*, and *name_of_institue*.	04/04/2022

	- Methods of *lab_assistant* class: __*init*__() method that gets invoked upon instantiation and takes values of class attributes. The *display*() method that prints class attribute values along with attributes of its super class. - Attributes of *office_assistant* class: *designation* = “Office Assistant” (static), *highest_qualification*, *year_of_joining*, and *name_of_institute*. - Methods of *office_assistant* class: __init__() method that gets invoked upon instantiation and takes values of class attributes. The display() method that prints class attribute values along with attributes of its super class. - Attributes of *peon* class: *job_role* = “office Peon” (static), *qualification*, *year_of_joining*, and *name_of_institute*. - Methods of *peon* class: __*init*__() method that gets invoked upon instantiation and takes values of class attributes. The *display*() method that prints class attribute values along with attributes of its super class.	
16	Create three classes named - *C, Python*, and *Web_Designing* each having two primary attributes as *learnings_* and *name_of_professor*. Derive a class named student from these classes. The student class hase following methods and attributes: I. Global std_college attribute with static values. II. *__init__()* method with attributes - *std_name*, *std_enrollment_no*, and *std_course*. III. *display()* method to display various attribute values of the terminal.	04/04/2022
17	Write a python program to demonstrate the use of data hiding.	18/04/2022
18	Write a python program to create a class named *area.* Define a class method *find_area()* that can find areas of different shapes whose value is given by the user. Invoke the class method by instantiation and prove method overloading.	18/04/2022
19	Write a python program to demonstrate the use of method overriding.	18/04/2022
20	Write a python program to demonstrate the use of try-catch block for exception handling.	18/04/2022
21	Write a python program to raise an exception with the python raise keyword.	25/04/2022
22	Write a python program to demonstrate the try-finally block.	25/04/2022
23	Write a python program to read the content of a file and return the number of words in a file to the user.	25/04/2022

24	Write a python program to read and show first N lines to the user. The number of lines N will be taken from user input.	02/05/2022
25	Write a python program that takes input of a student course from the user and and write it in a file.	02/05/2022
26	Write a python program to copy the content of one file to another file.	09/05/2022
27	Write a python program that creates 26 text files named A.txt, B.txt, and up to Z.txt.	09/05/2022

Practical 1

Write a python program to greet users with welcome messages using print() method.

Practical 2

Write a python program to demonstrate the creation of List data structure along with its methods - append(), extend(), insert(), remove(), clear(), index(), count(), sort(), reverse(), and copy().

a. Demonstrate positive and negative indexing with python List.

b. Demonstrate slicing operations on python List.

c. Demonstrate updation on List elements in python.

d. Demonstrate deletion of a single python list element and multiple elements using slicing operator..

Practical 3

Write a python program to demonstrate the creation of tuples along with its methods - count() and index().

a. Demonstrate positive and negative indexing with python Tuple.

b. Demonstrate slicing operations on python Tuple.

c. Demonstrate updation on Tuple elements in python.

Practical 4

Write a python program to demonstrate the creation of a Dictionary student with the name, age and branch of a student.

a. Demonstrate the updation of python dictionary.

b. Demonstrate the removal of elements from the python dictionary.

c. Demonstrate the use of following dictionary methods - clear(), copy(), get(), items(), keys(), popitem(), and values().

Practical 5

Demonstrate the use of basic string methods in python - lower(), upper(), join(), split(),find(), and replace().

Practical 6

Write a python program to implement basic arithmetic operations on user entered numbers.

Practical 7

Write a python program to count how many times a specific number is occurring in a list. Take user input for both numbers and a list.

Practical 8

Write a python program that takes N number of integers from the user in a python list. Create a function that takes the list of user entered numbers and returns MAX and MIN numbers from that list to the user.

Practical 9

Write a python program to perform basic matrix operations on user entered matrices.

Practical 10

Write a python program to perform basic operations of a calculator. Provide choice for operations to users and make a program iterative. Provide specific exit option to users.

Practical 11

Write a python program to demonstrate the use of arbitrary arguments.

Practical 12

Create a class named student having attributes - std_name, std_age, std_branch, and std_city. Create a method named get_data() in student class that takes user input for these attributes and a method named display() that prints the attribute values on the terminal. Call both the methods by creating an instance std_obj of the class student.

Practical 13

Write a python program to demonstrate basic banking operations. Create a class named banking having separate class methods for each operation. Call each method with an instance of the class and attribute values to be taken from the user.

Practical 14

Create a class named employee having attributes - emp_name, emp_age, and emp_city. Create a method named get_data() in employee class that takes user input for these attributes. Derive a class named emp_derived() from the employee class, having an __init__() method that displays the attributes of the employee class upon instantiation.

Practical 15

Create a base class named university with its attributes - name, year_of_estd, and city. Derive following class from the super class university: professor, lab_assistant, office_assistant, and peon. Make the program choice based for user. The attributes and method of various class are as below:

- Attributes of professor class: designation, highest_qualification, area_of_research, year_of_joining, year_of_ experience, and name_of_institute.

- Methods of professor class: __init__() method that gets invoked upon instantiation and takes values of class attributes. The display() method that prints class attribute values along with attributes of its super class.

- Attributes of lab_assistant class: designation = “Lab Assistant” (static), highest_qualification, additiobnal_skilss, year_of_joining, and name_of_institue.

- Methods of lab_assistant class: __init__() method that gets invoked upon instantiation and takes values of class attributes. The display() method that prints class attribute values along with attributes of its super class.

- Attributes of office_assistant class: designation = “Office Assistant” (static), highest_qualification, year_of_joining, and name_of_institute. - Methods of office_assistant class: __init__() method that gets invoked upon instantiation and takes values of class attributes. The display() method that prints class attribute values along with attributes of its super class.

- Attributes of peon class: job_role = “office Peon” (static), qualification, year_of_joining, and name_of_institute.

- Methods of peon class: __init__() method that gets invoked upon instantiation and takes values of class attributes. The display() method that prints class attribute values along with attributes of its super class.

Practical 16

Create three classes named - *C*, *Python*, and *Web_Designing* each having two primary attributes as *learnings_* and *name_of_professor*. Derive a class named student from these classes. The student class hase following methods and attributes:

1. **Global std_college attribute with static values.**
2. ***__init__()* method with attributes - *std_name*, *std_enrollment_no*, and *std_course*.**
3. ***display()* method to display various attribute values of the terminal.**

Practical 17

Write a python program to demonstrate the use of data hiding.

Practical 18

Write a python program to create a class named *area.* Define a class method *find_area()* that can find areas of different shapes whose value is given by the user. Invoke the class method by instantiation and prove method overloading.

Practical 19

Write a python program to demonstrate the use of method overriding.

Practical 20

Write a python program to demonstrate the use of try-catch block for exception handling.

Practical 21

Write a python program to raise an exception with the python raise keyword.

Practical 22

Write a python program to demonstrate the try-finally block.

Practical 23

Write a python program to read the content of a file and return the number of words in a file to the user.

Practical 24

Write a python program to read and show first N lines to the user. The number of lines N will be taken from user input.

Practical 25

Write a python program that takes input of a student course from the user and and write it in a file.

Practical 26

Write a python program to copy the content of one file to another file.

Practical 27

Write a python program that creates 26 text files named A.txt, B.txt, and up to Z.txt.