



SCHOOL OF ENGINEERING AND TECHNOLOGY

Record of Applied and Action Learning (Programming Practice)

Subject Name: Python Programming

Subject Code: CUREL -1008

Semester: IST

Name: K. Abhinav

Registration No.: 241314100006

Program/Branch: BCA

Specialization: BT ECH

Academic Year: 2014 - 15

Campus: Vizianagaram



Centurion
UNIVERSITY

**CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT
ANDHRA PRADESH**

Certificate

This is to certify that Mr./Ms. K. Abhinav having
Registration No. 2U114100006 of 1st Semester,
..... B-TECH R.C.A Program, SOET
School, Uttamagaram Campus has completed
number of experiments in Applied and Action
Learning Laboratory and fulfils the course
requirements.

Signature of the HoD/Dean

Signature of the Faculty

Office Seal

INDEX

Sl. No.	Date	Name of the Experiment	Page No.	Remark	Faculty Signature
1)		Introduction to Python and setting up the environment	1		
2)		Variables Data Types and Type Conversion	3		
3)		Operators in Python	5		
4)		working with Strings	7		
5)		understanding lists and list operations	9		
6)		conditional statements	11		
7)		loops in Python and cool control	13		
8)		Defining functions in Python and function arguments	15		
9)		Introduction to Python data structures	17		
10)		working with Functions in python	19		
11)		classes and objects in Python	21		
12)		Inheritance and Polymorphism in Python	23		
13)		Encapsulation in Python and private variables	25		
14)		file handling in Python	27		
15)		Modules and packages in Python	29		
16)		Introduction to Matplotlib lib and Basic plotting	31		
17)		Creating line Plot and Scatter plots with matplotlib	33		
18)		Customizing plots	35		
19)		Plotting with Pandas and Matplotlib lib integration	37		
20)		Introduction to Plotly for interactive visualization	39		



School: S.O.E.T 2024-25 Campus: Mizianagaram
Academic Year: 1st Subject Name: Python Program Subject Code: CW.BC - 1008
Semester: 1st Program: B.C.A Branch: B.TECH Specialization:

Date:

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Introduction to Python and setting up the environment
Learning Outcome:

Concepts learned (Mention 2/3 principles):

- 1) Installation and setup: Ensure python and essential libraries are installed correctly along with a suitable IDE for efficient coding.
- 2) Basic understanding and practice: Learn python's fundamental features and start with simple scripts to get comfortable with the language.

New techniques learned:

- 1) system compatibility: Ensure your computer system meets the requirements for installing python
- 2) write and run basic scripts: Start with simple python scripts to familiarize yourself with the language syntax and execution.

* Related Project/Practice work experienced and learned:

Classroom

i) Practicework

- 1) Download the latest process and follow the installation.
 - 2) Choose and install Integrated development environment.
 - 3) write a simple script to print "Hello world" to the screen.
- 2) Learned: 1) Gained familiarity with Python syntax, similar scripts 2) explore standard libraries to perform tasks and build foundation skills.

* New Software/Machine/Tool/Equipment/Experiment learned:

- 1) Python interpreter: the core software need to run and execute the Python code. Download from Python website.
- 2) Integrated Development Environment: Tool to edit and deploy code efficiently like Jupyter notebook, vs code, Pycharm.

* Application of concept(s) (preferably real life scenario):

Data Analysis for small business.

- 1) Install Python and Jupyter notebook
- 2) Prepare data
- 3) Load analyze data in Jupyter notebook
- 4) Make data - driven decisions.

* Case Studies/Examples:

1) Print ("Hello world") Output Hello world

2) $a = 5$

$b = 3$

print(a+b) Output: 8

Assessment:

Marks Obtained: 10 / 10

Signature of the Student: Abhi

Name: Ic. Abhinav

Regn. No.: 241814100006

Page No. 4
As applicable according to
One sheet per page (10-20)

Signature of the Faculty:



School: SOFT

Campus: Vizianagaram

Academic Year: 2024-25 Subject Name: Python Programming Subject Code: CUBC-1008

Semester: 1st Program: BCA Branch: BTech Specialization:

Date:

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Variables, Data type, conversion.

Learning Outcome:

Concepts learned (Mention 2/3 principles):

- 1) Understanding Data types:
 - 1) variables: contains that stored values.
 - 2) Data types: 1) integer (int): whole numbers 2) float: Numbers with a decimal point 3) string (str): A sequence of characters 5) Boolean (Bool): Represents True or False
 - 3) Type conversion: 1) Implicit conversion: Python automatically converts the datatype to another, when appropriate.

New techniques learned:

- 1) Variable declaration: Assign values to variables and understand their data types.
- 2) Identify and utilize datatypes: Recognize the properties and uses of different data types (int, float, etc.)
- 3) Mathematical operations: perform arithmetic operations on integers and float
- 4) String Manipulation: use string methods to modify and handle text data.

* Related Project/Practice work experienced and learned:

- 1) Practice work: 1) Define and use variables: practice variables of different data types 2) Implicit conversion: practice operations that involve type conversion 3) Convert variables from one type to another using different type casting
 - 2) Learned: Recognize the use of different type data in Python 2) Implicit conversion: understand how automatically converts data types during operation
- * New Software/Machine/Tool/Equipment/Experiment learned:

1) Python interpreter: The core software needed to execute Python code. Download from the official Python. 2) Integrated Development Environment (IDE) to write, edit and debug code efficiently like Jupyter notebook vs code PyCharm.

* Application of concept(s) (preferably real life scenario):

By defining variables using appropriate data and performing type conversions you can a simple budget tracking application that helps you to manage your monthly expenses effectively.

age = 25

- * Case Studies/Examples:
- 1) Variables of data types: Int print(age) H output age
 - 2) float height = 5.9 3) String name = "yash" print(name)
 - a) Boolean is used = True
print(c is - student) H output true

Assessment:

Marks Obtained: 10 / 10

Signature of the Faculty:



Signature of the Student: 

Name: K. Abhinav

Regn. No.: 241814100006

* As applicable according
One sheet per topic



School: SOE..... Campus: Vizianagaram
Academic Year: 2014-15 Subject Name: Python Programming Subject Code: C.U.B.C-1003
Semester: 1st Program: BCA Branch: BTech - Specialization: _____

Date: _____

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Operators in Python.

Learning Outcome:

Concepts learned (Mention 2/3 principles):

By using operators we learned.

- 1) Arithmetic operators: perform basic math functions using arithmetic operators
- 2) Comparison operators: compare values to check conditions using comparison operators.
- 3) Operators are essential for making decision and performing calculations in Python

Programmes: -

New techniques learned:

- 1) Combining operations for complex condition:
* use arithmetic operator within comparison expression to evaluate complex conditions.
- 2) Chaining comparison operator:
* combine multiple comparison operators to create clear or concise conditions.
- 3) Using logical operator for compound condition * logical Operator like (and, or, not)

* Related Project/Practice work experienced and learned:

- 1) Practice works:
 - 1) Basic Arithmetic operations: perform subtraction, multiplication, division, modules
 - 2) Comparison operators: compare values using operators
 - 3) Logical operators: use logical operators to combine conditions
- 2) Learned: used Arithmetic and comparison operators together to evaluate more interactive conditions, multiple conditions using logical operators (and, or, not)
- * New Software/Machine/Tool/Equipment/Experiment learned:

* New Software/Machine/Tool/Equipment/Experiment learned:

- 1) python interpreter: The core software needed to execute python code. Download from the official website.
- 2) Integrated Development Environment: Tools to edit and debug code efficiently like Jupyter notebook vs Code PyCharm.

* Application of concept(s) (preferably real life scenario):

If we want to develop a simple fitness application to monitor your daily physical activity metrics Python arithmetic, comparison operators, and loops are essential for calculating and evaluating different aspects of your fitness data.

* Case Studies/Examples:

1) Arithmetic operator : Add(t) a=5
b=3
print(result)
8.

3) multiplication $a=7$
 $b=2$
 result = a * b
 print(result) 4

Assessment:

Marks Obtained: / 10

Signature of the Student: Abhi

Name: K. Abhiram

Regn. No.: 201814100006

Signature of the Faculty: 

*As applicable according
to relevant legislation



School: <u>" SOET</u>	Campus: <u>Vijiana garam</u>		
Academic Year: <u>2014-15</u>	Subject Name: <u>Computer Programming</u>	Subject Code: <u>L018</u>	
Semester: <u>1st</u>	Program: <u>BTech</u>	Branch: <u>BCA</u>	Specialization: _____
Date: _____	Classroom Learning		
* (Learning by Listening and Observations)			

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: → work with strings

Learning Outcome:

Concepts learned (Mention 2/3 principles):

- 1) String concatenation and slicing:
 - 1) Concatenation (combining two or more strings into one)
 - 2) slicing (extracting a portion of a string using index ranges)
- 2) String methods:
 - 1) upper(): converts all characters into string to uppercase
 - 2) lower(): converts all characters into a string to lowercase
 - 3) find(substring): returns the index of first occurrence of the substring
 - 4) Replace old, new: Replaces occurrences of a substring with another substring

New techniques learned:

- 1) String concatenation:
 - + combine two or more strings to form a single string
- 2) String slicing:
 - + extract a portion of string using index ranges
- 3) Using String methods:
 - + changing case: convert strings to upper and lower case
 - 4) finding substring: locate the position of substring with ^{String}
 - 5) splitting strings: divide a string into a list of substrings based on a delimiter

F
P
S
C
Z
* Related Project/Practice work experienced and learned:
1) Practice work: combine strings
2) find substring
Text = "python's fun"
index = text.find("fun")
print(index) # output 10

2) Learned 1) string concatenation: combine multiple strings into a single string 2) string slicing: extracting parts from a single string

* New Software/Machine/Tool/Equipment/Experiment learned:

1) Python interpreter: The core software needed to execute python code. Download from the official website.

2) Integrated development environment/IDE: Tools to write, test, and debug code efficiently.

like Jupyter notebook, VS code, pyCharm

* Application of concept(s) (preferably real life scenario):

* Creating a personalized greeting cards
we want to create a personalized greeting card for friends and family using Python we can use slicing and methods to customize our code.

* Case Studies/Examples:

1) concatenation: str1 = "Hello" 2) slicing: str2 = "world"
str1 + str2
result = str1 + str2
print(result) # output "HelloWorld"

3) change case: text = "Python" 4) text:
upper_text = text.upper() index:
print(upper_text) # output "Python" print(index)

Signature of the Student:
Name: K. Abhirav
Regn. No.: 201814000

As
Ma
Assessment:
Marks Obtained: 10 / 10

Sig

Signature of the Faculty:



School:	SOET	Campus:	Villianagaram
Academic Year:	20-25	Subject Name:	DAUP
Semester:	1 st	Subject Code:	Lo18
Program:	B.TECH	Branch:	BCA
Date:			
Classroom Learning (Learning by Listening and Observations)			

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Understanding List operations

Learning Outcome:

Concepts learned (Mention 2/3 principles):

- 1) creating multiple lists: ordered collection of items which can be different data types
2) Accessing elements using indices to access list items.
3) common list operations: 1) Appending item Adding an item to the end of list 2) inserting items Adding an item at a specify index 3) Removing items removing item from the list 4) popping items Removing and returning the item at specify index 5) starting list starting items in ascending order.

New techniques learned:

Creating and Accessing lists: 1) Create a list Define a list to store multiple items 2) Access list elements Use indices to access specify items in a list

2) Manipulating lists 1) Appending items Add a new item to the end of the list 2) inserting items insert an item at a specify index in the list 3) Removing items a specify item from list 4) popping items and remove and return item at specify index 5) starting lists: Start the list in ASD order or DS order.

* Related Project/Practice work experienced and learned:

- 1) Practice work: 1) Create list = fruits ["apple", "banana"]
- 2) Access elements: first fruits [0] second fruits - fruit [1]
- 3) Appending items: print (first-fruit) [1] # output fruits.append ("orange")
- 3) Appending items: print fruits # output "apple", "banana", "orange"
- 2) Learned: 1) lists can store multiple items of different types 2) use indices to access element in list

* New Software/Machine/Tool/Equipment/Experiment learned:

- 1) Python interpreter: The core software needed to execute python code. Download from the official website.
- 2) Integrated development environment (IDE): write, edit and debug code efficiently, like Jupyter notebook, VS code, PyCharm.

* Application of concept(s) (preferably real life scenario):

+ Grocery shopping list:

We need to manage grocery shopping list by adding items you think removing item when they are purchased and organizing list to make shopping more efficient.

* Case Studies/Examples:

- 1) Creating list: fruits ["apple", "banana", "cherry"]
Print (fruits) # output ["apple", "banana", "cherry"]
- 2) Inserting: fruits.insert (1, "blue berry")
Print (fruits) # output ["apple", "blue berry", "banana", "cherry"]
- 3) Remove: fruits.remove ("banana")
Print (fruits) # output ["apple", "blue berry", "cherry"]

Assessment:

Marks Obtained: 10/10

Signature of the Student: Atish

Name: K. Abhinav

Regn. No.: QUILS1410000

Signature of the Faculty:



School: SOET Campus: Villianagaram

Academic Year: 2025 Subject Name: DAVP Subject Code: 1018

Semester: 1st Program: B TECH Branch: BCA Specialization: _____

Date: _____

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Conditional statements

Learning Outcome:

Concepts learned (Mention 2/3 principles):

Control flow with conditions: Conditional statements are used to control the flow of a program based on a certain condition.

Logical expressions: Conditional statement rely on logical expressions that evaluate to either true or false.

Expressions: Comparison operator, logical operator

New techniques learned:

Decision-making with logical conditions

Conditional statements allow program based on a certain logical conditions if conditional statement like if, elif, and else statements.

Combining and evaluating conditions

Using logical and comparison operators you can combine and evaluate conditions to control the program's flow.

* Related Project/Practice work experienced and learned:

Practice work: conditional statement
what we've learned about conditional
else) Simple temperature check Grade equal

Learned: 1) control flow with programs based on
provided 2) logical expressions how to use
and logical operators to form conditions
which block of code will execute

* New Software/Machine/Tool/Equipment/Experiment learned:

1) python interpreter: The core software needed
and execute python code Download from the
2) Integrated Development Environment (IDE): Tools
edit and debug code efficiently.
like Jupyter notebook vs code pyth

* Application of concept(s) (preferably real life scenario):
conditional statements are incredibly useful
because they allow us to make decisions
conditions like Traffic light system

Imagine you are programming a traffic light
you need the traffic lights to change based on

* Case Studies/Examples:

Traffic light system

time=15

```
if time == 10
    print("Green light")
elif time == 13
    print("Yellow light")
else
    print("Red light")
```

Assessment:

Marks Obtained: 9/10

Signature of the Student

Name: K. Abhinav

Regn. No.: 2U181410002

Signature of the Faculty:



School: SOET Campus: Villanagaram
Academic Year: 24-25 Subject Name: DAVF Subject Code: 1D 18
Semester: 1st Program: B.TECH Branch: BCA Specialization:
Date: _____

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Loops in python and loop control

Learning Outcome:

Concepts learned (Mention 2/3 principles):

1) Iteration and Repetition

Loops are fundamental concept in programming that allow for the repetition of a block of code multiple times like for while

2) Loop control statements change the execution flow of the loop from its normal sequence like break and continue

New techniques learned:

1) Technique for using loops : i) iterating over a sequence with for loops techniques use for loops to iterate over each element into a sequence (like list tuple or string)

ii) Repeating tasks with while loop

3) Techniques for control loop exiting a loop early with break use the break statement to exit a loop

4) Skipping iterations with continue use the continue statement to skip the current iterations to move the next to a new when condition is met.

* Related Project/Practice work experienced and learned:

- 1) Practicework, sum of the numbers writing a program to calculate the sum of the first 10 natural numbers.
- 2) Countdown: write a program that counts down from 10 to 0. Learned: i) iteration and repetition we practiced for and while loops to repeat tasks and over sequence.

* New Software/Machine/Tool/Equipment/Experiment learned:

1) Python interpreter! The core software needed to write and execute Python code. Download official Python website. 2) Integrated development tool to write, edit and de bug, code efficiently like Jupyter notebook vs pyCharm.

* Application of concept(s) (preferably real life scenario):

OOPS and loops controls are essential in scenarios where repetitive tasks needed to be performed. decisions must be based on certain conditions like monitoring system resources using while loop continuously and break to exit the loop.

* Case Studies/Examples:

Monitoring system resources

Important random code:	<pre> if CPU_usage > print warning usage is too high break </pre>
<pre> CPU_usage = 0 while CPU_usage < 90 CPU_usage = random(0,100) print(f"current CPU usage is {CPU_usage}") </pre>	

Assessment:

Marks Obtained: / 10

Signature of the Faculty:

Signature of the Student: Abhishek

Name: Rb K. Abhishek

Regn. No.: 2U181410000 6

School: SOET Campus: Villianogram

Academic Year: 2021-22 Subject Name: DAVP Subject Code: 1013

Semester: 1st Program: BTech Branch: BCA Specialization:

Date: _____

Classroom Learning

(Learning by Listening and Observations)

: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as 'why, when and how (application)'. Skill gain will be reflected through an answer to the question "how to do it". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Defining functions in python by and
Learning Outcome: function Arguments.

Concepts learned (Mention 2/3 principles):

Defining functions:

functions are reusable blocks of code designed to perform specific task using the def keyword naming the function specifying parameters 1) writing the function body 2) Returning value
Learn function Arguments

function can accept different types of argument to make them more flexible and dynamic 1) Positional Arguments
2) Keyword Argument 3) Default Argument 4) Arbitrary argument

Techniques learned:

Basic function definition: concept : Define a function using the def keyword followed by the function name

1) How to use: write the code you want to be executed inside the function body properly indented

2) Using positional and keyword Arguments: concept
Positional Arguments are passed to the function in order they are defined
Key word Arguments are passed with a key value pair

3) How to use: call the function by specifying argument in the required data

* Related Project/Practice work experienced and learned:

- 1) Practicework: Defining functions and using arguments simplegreeting function: cutting a function to print a person's name as an argument and print Personalize greeting
- 2) Learned: function definition, function arguments, Return statement, Arbitrary Arguments.

* New Software/Machine/Tool/Equipment/Experiment learned:

- 1) Python interpreter: The core software needed to run and execute python code. Download from the official website.
- 2) Integrated Development Environment: Tool to write and debug code efficiently like Jupyter book vs code Pycharm

* Application of concept(s) (preferably real life scenario):

functions and function Argument are invaluable in programming that can streamline and simplify like automatically daily tasks. Imagine you have many tasks you need to complete each day. You define a function to automate sending reminder tasks.

* Case Studies/Examples:

Automating Daily tasks

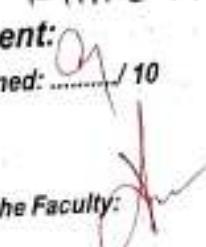
```
def send_remainder(task, time = "8:00 AM"):
    print(f'{task} at {time}'
```

tasks = ["Meeting with team", "Doctor appointment"]

time = ["10:00 AM", "1:00 PM", "6:00 PM"]

for i, time in zip(tasks, time):

Signature of the Student: Abhinav

Assessment: 
Marks Obtained: 9/10

Name: K. Abhinav

Regn. No.: Q41814100006

Signature of the Faculty: 



School: SOFT Campus: VILVANA.GOV.IN

Academic Year: 24-25 Subject Name: DAVP Subject Code: L01A

Semester: 1ST Program: BTech Branch: BCA Specialization:

Date:

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Introduction to Python Data Structures

Learning Outcome:

Concepts learned (Mention 2/3 principles):

- 1) Understanding different data structures: lists ordered mutable collection of items of ten off the same type
 - 2) Tuples ordered immutable collection of items 3) Diction^{aries} collections of key value pairs where each key is unique
 - 4) sets uncleaned collections of unique items.
- 2) Operations and methods: lists Appending removing
- 2) Tuples : Accessing element slicing 3) Dictionaries: Accessing adding updating sets Adding removing items.

New techniques learned:

- 1) Core techniques for using data structures:
 - 1) Appending and removing elements Add items using append()
 - 2) slicing : extract sublists using the slice notation start end step
 - 3) Tuples : Accessing elements retrieve items by index directly to lists
 - 4) unpacking: Assign tuple elements to variables
- 2) Advanced techniques for data structure
 - 1) Accessing and modify values: use keys to access values
 - 2) Iteration over items use loops to iterate over dictionary key values or both.

* Related Project/Practice work experienced and learned:

Practice work: 1) Lists: Create a list of your fav.
 2) Tuples: Create a tuple with the different colors
 perform the following operations 3) Dictionaries:
 dictionary with details of a person 4) Create
 unique numbers and perform
 learned: 1) Lists: ordered collection that allow
 tuples: immutable ordered collection ordered col.

* New Software/Machine/Tool/Equipment/Experiment learned:

- 1) Python interpreter: The core software needed to execute python code Download from the official
- 2) Integrated Development Environment: To write and debug code efficiently like Jupyter book vs code PyCharm

* Application of concept(s) (preferably real life scenario):

Data structure list

when you needed to organize your weekly grocery shopping a list can you keep track of items.

* Case Studies/Examples:

```
shopping list: shoppinglist = ["Milk", "bread"]
shoppinglist.append("butter")
shoppinglist.remove("bread")
print(shoppinglist)
#output
['Milk', 'eggs']
```

Assessment:

Marks Obtained: / 10

Signature of the Faculty: 

Signature of the Student: Abhi

Name: K. Abhinav

Regn. No.: 241814100006

Page

*As applicable according to
One sheet per topic (M-S)



School: <u>SOET</u>	Campus: <u>Villianagarom</u>		
Academic Year: <u>2015</u>	Subject Name: <u>DAV</u>	Subject Code: <u>1018</u>	
Semester: <u>3rd</u>	Program: <u>BT ECT</u>	Branch: <u>BCA</u>	Specialization: _____
Date: _____	Classroom Learning (Learning by Listening and Observations)		

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: working with functions in python

Learning Outcome: :-

Concepts learned (Mention 2/3 principles):

- 1) Recursion: It is a technique where a function calls itself in order to solve smaller instance of same problem like Base case and Recursive case
- 2) Lambda function: It is also known as anonymous functions are small unnamed functions defined using the lambda keyword. They are often used for programming contexts like syntax and simplicity and limited scope.

New techniques learned:

- 1) Recursion Techniques: Base case identify the simple case that can be solved directly without further recursion. Prevents infinite recursion and ensure function terminates. Lambda function technique: compact syntax use the keyboard to define simple one-linear functions without the need for a formal function declaration. Function programming integration: utilize lambda() within high order functions like map(), filter() for concise and readable code.

* Related Project/Practice work experienced and learned:

- 1) Practice works
 - 2) Base case identification: the base case is when itself within "n-1" progressively working toward
 - 3) Learned: Lambda function Gain proficiency in functions for concise operations Reorganize their functional programming and their ensuring in right
- * New Software/Machine/Tool/Equipment/Experiment learned:

Python interpreter! The core software needed to execute Python code Download from the official website 2) Integrated Development Environment write, edit and debug like code efficiently Use Jupyter notebook vs code pycharm

* Application of concept(s) (preferably real life scenario):

Starting complex data

Consider we have a list of dictionaries in employee records and you want to sort them by age without defining a full function.

* Case Studies/Examples:

```
employees: [ { "name": "Alice", "age": 30 },
              { "name": "Bob", "age": 25 },
              { "name": "Charlie", "age": 35 } ]
sorted_employees = sorted(employees, key=lambda x: x["age"])
for employee in sorted_employees:
    print(employee)
```

Assessment:

Marks Obtained: 9 / 10

Signature of the Faculty:

Signature of the Student: Abhi

Name: K. Abhinav

Regn. No.: 241814100006



School: SOCT Campus: Villanagaram

Academic Year: 21-25 Subject Name: DAuP Subject Code: LO18

Semester: 5th Program: BTech Branch: BCA Specialization:

Date:

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Classes and object in Python

Learning Outcome:

Concepts learned (Mention 2/3 principles):

Encapsulation: It is the practice of bundling data and methods that operate on the data into a single unit called a class.

a) Inheritance: It is a mechanism where a new class inherits the attributes and methods of an existing class. This allows for code reusability and the creation of hierarchical relationship between classes.

New techniques learned:

i) Defining class and objects: Encapsulation starts with defining classes and creating objects. A class serves as a blueprint for objects bundling data and methods together.

ii) Creating sub classes: Inheritance allows a class to inherit attributes and methods from another class. The super() function is used to call the superclass's __init__ method to ensure proper initialization.

* Related Project/Practice work experienced and learned:

Practices work: 1) Defining classes and objects: Program a class with attribute name 'analoge' and basic() and between human - age().
 2) Learned: Inheritance allows us to build existing classes prompting code reuse and creating a relation between classes.

* New Software/Machine/Tool/Equipment/Experiment learned:

1) Python interpreter: The core software needed to execute python code. Download from Python website 2) Integrated Development Environment to write edit and debug code efficiently - Jupyter notebook vs code pycharm

* Application of concept(s) (preferably real life scenario):

Let's consider a simple real life scenario where you might use classes and objects in Python managing a collection of movies in a personal movie library.

* Case Studies/Examples:

Movie library: Class movie:

```
def __init__(self, title, director, year):
    self.title = title
    self.direction = direction
    self.year = year
    def get_details(self):
        return f'{title} | {director}, {year}'
```

Assessment:

Marks Obtained: 9 / 10

Signature of the Student: Abhi

Name: K. Abhinav

Regn. No.: 24181410006

Signature of the Faculty:



School: SO ET Campus: Vilianagar

Academic Year: 14-15 Subject Name: DAVP Subject Code: 1018

Semester: 1st Program: DTech Branch: BCA Specialization:

Date:

Classroom Learning

(Learning by Listening and Observations)

e: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Inheritance and Polymorphism in Python

Learning Outcome:

Concepts learned (Mention 2/3 principles):

1) Inheritance: It is a fundamental OOP principle where a new class (subclass) inherits attributes and methods from an existing class (superclass).

2) Polymorphism: It allows methods to perform differently based on the object calling them. It enables a single function to handle different types of objects.

New techniques learned:

1) Creating Subclasses: Inheritance allows the creation of substance that inherit attributes and method from a superclass. This helps in creating a hierarchical relationship and reusing codes.

2) Extending functionality: Substances can add a new attributes and methods or existing ones to provide specialised behaviour while still leaving ragging the common functionality of the Super class.

* Related Project/Practice work experienced and learned:

- 1) practice work: Implementing a base class to create a base class vehicle with a method. Create subclasses car and Bike that inherit vehicle and override the move() method to specify.
- 2) learned: Polymorphism provides flexibility allowing objects of different classes to be treated as instant.

* New Software/Machine/Tool/Equipment/Experiment learned:

- 1) Python interpreter: The core software needed to run and execute Python code. Download from the Python website. Integrated development environment to write, edit and debug code efficiently like Jupyter notebook, VS Code, PyCharm.

* Application of concept(s) (preferably real life scenario):

Let us consider a simple real life scenario where you might classes and objects in managing a collection of movies in a personal movie library.

* Case Studies/Examples:

Different types of Animal

```
Class Animal
def __init__(self, name):
    self.name = name
    self.sound = "Some generic animal sound"
print(self.name + " makes a " + self.sound)
```

```
Class Dog(Animal)
def make_sound(self):
    return "Bark"
Class Cat(Animal)
def make_sound():
    return "Meow"
```

```
animal = dog
dog = Dog()
cat = cat
animal = cat
animal = dog
animal = Cat()
```

Assessment:

Marks Obtained: 9/10

Signature of the Student: Abhi

Name: K. Abhinav

Regn. No.: JU1814100006

Signature of the Faculty:



School: SO&T Campus: Villianogram
Academic Year: 2014-15 Subject Name: DAWP Subject Code: 1013
Semester: 3rd Program: BTech Branch: BCA Specialization:

Date:

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Encapsulation in Python and Private Variables
Learning Outcome:

Concepts learned (Mention 2/3 principles):

1) private variables: Encapsulation is often implemented using private variables. These are variables that are meant to be accessed within the class then they are definition getter and setter methods. To control access to private variables publicly to load or modify private variables in a controlled way. By using getter and setter you can add logic enforce constants radiation when the variables value is accessed or changed.

New techniques learned:

- 1) Restricting: Access with private numbers
- 2) private variables: Although Python doesn't enforce true private variables you can indicate that a variable should be private and prefixing its name with a single underscore (-)
- 3) using public methods for controlled Access these variability this allows you to add logic or validation transformation or constraints when these variables are accessed.

* Related Project/Practice work experienced and learned:

1) Practice work:

Define a class Bank account with private
for account balance and account holders,
public methods to deposite and withdraw,
to check the balance 2) Learned: 1) Private var
prefixing variables names with __ to indicate
they are should not accessed directly from
outside the class.

* New Software/Machine/Tool/Equipment/Experiment learned:

1) Python interpreter: The core software needed
to write and execute python code. Download from
official python website 2) Integrated development
environment: To write, edit and debug
efficiently like Jupyter notebook or Python IDLE.

* Application of concept(s) (preferably real life scenario):

Library system: In a library each book has
title, author and status. The status
should be private, ensure it can only change
through controlled methods.

* Case Studies/Examples:

Bank accounts

```
class BankAccount:
    def __init__(self, initial_balance):
        self.balance = initial_balance
```

```
def deposit(self, amount):
    if amount > 0:
        self.__balance += amount
    else:
        raise ValueError("Amount must be positive")
```

Assessment:

Marks Obtained: 9/10

Signature of the Student: Abhi

Name: K. Abhinav

Regn. No.: 241814100006

Signature of the Faculty:

* As applicable according
One sheet per topic



School: S.O.T Campus: Villanagaram
Academic Year: 24-25 Subject Name: P.A.P. Subject Code: 1018
Semester: 1st Program: BTech Branch: BCA Specialization:

Date:

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as 'why, when and where (application)'. Skill gain will be reflected through an answer to the question 'how to do things'. Attitude change will be observed through visible alteration in behavior.

Name of the Topic: file handling in python

Learning Outcome:

Concepts learned (Mention 2/3 principles):

- 1) opening and closing file: 1) opening file to work with a file the first step is to open it using open()
- 2) closing a file: After completing the operation on the file its essential to close it using the close() method to free up system resources.
- 3) Recycle and writing to files: Reading you can read to content of a file using methods read(), readline() and readlines()

New techniques learned:

- 1) opening and closing files before performing any operations on a file you need to open it similarly you should close the file after completing your operations to free up system using open(), close()
- 2) Reading and writing to files: Files operation often involve reading data from or writing files python provides unreal methods for reading from and writing to files Techniques

* Related Project/Practice work experienced and learned:

1) Practice work: 1) Appending to a file: write script to append new content to output, the updated content on the file.
writing a file: write a python script to new text named output.txt writeson to the file and close it.

* New Software/Machine/Tool/Equipment/Experiment learned:

1) Python interpreter: The core software needed to write and execute python code. Download from official python website. 2) Integrated Development Environment: to write, edit and debug efficiently like Jupyter book vs code pycharm.

* Application of concept(s) (preferably real life scenario):

Student management system: In student management system we need to store student name, their grades. We will use file handling to manage this data allowing us to read and write to a file.

* Case Studies/Examples:

writing student record to a file
student - s
def write_student_records
students with open filenames
for name grade in student
file write (f, name, grade)

student - s
"Alice": "A"
"Bob": "B"
"Charlie": "A"
David": "C"
write student records text

Assessment:

Marks Obtained: / 10

Signature of the Faculty:

Signature of the Student: Abhi

Name: K. Abhinav

Regn. No.: 241814100006

School: SO-E-T Campus: M.I.T. Bangalore

Academic Year: 2014-15 Subject Name: P.A.U.P Subject Code: 1013

Semester: IST Program: DTE-LIT Branch: BCA Specialization:

Date:

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as 'why, when and where (application)'. Skill gain will be reflected through an answer to the question 'how to do things'. Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Modules and Packages in Python.

Learning Outcome:

Concepts learned (Mention 2/3 principles):

- 1) Creating and using modules: A module is simply a file containing Python code. It can define functions, classes and variables that can be imported and used.
- 2) Creating and using packages: A package is a way of organizing multiple modules into a directory. A package contains __init__.py file which can be empty or execute initialization code for the package.

New techniques learned:

- 1) Creating a module: Save your functions, classes or variables in a py file for ex. Create a file named mymodule.py
- 2) Create a package: Organize your modules in a directory structure with an __init__.py file.
- 3) Using a package: Import the package and its modules in another script using the form import statement.

- * Related Project/Practice work experienced and learned:
- 1) Pracitework. 1) Create a directory named Classmate with an __init__.py file and multiple .py files.
 - 2) Define functions in arithmetic.py and calculate.py.
 - 3) Use the package in main.py.
 - 4) Learned: creating modules and individual .py files that contain python code such as calculator.py.

- * New Software/Machine/Tool/Equipment/Experiment learned:
- 1) Python interpreter: The core software needed to write and execute python code. Download from official python website.
 - 2) Integrated Development Environment: To write, edit and debug efficiently like Jupyter notebook vs code.

* Application of concept(s) (preferably real life scenario):

School management system:

Modules will create separate modules for students, teachers and courses.

Packages will organize these modules.

Package named school management.

* Case Studies/Examples:

```
student.py: class student:  
    def __init__(self, name, age, student_id):  
        self.name = name  
        self.age = age  
        self.student_id = student_id
```

Assessment:

Marks Obtained: 9 / 10

Signature of the Student: Abhinav

Name: K. Abhinav

Regn. No.: 201814100006

Signature of the Faculty:

* As applicable according
One where applicable



School: ... School ... Campus: ... Williamagonam

Academic Year: 2021-22 Subject Name: D.A.U.P. Subject Code: 101A

Semester: 1 Program: B.TECH Branch: BCA Specialization: ...

Date: ... Classroom Learning
[Learning by Listening and Observations]

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as 'why, when and where (application)'. Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Introduction to Matplotlib and Basic Plotting techniques.

Learning Outcome: Concepts learned (Mention 2/3 principles):

1) Importing and setting up matplotlib: matplotlib is a comprehensive library for creating static, Animate and interactive visualization in python.

2) Customizing and enhancing plots: Beyond simple plots, matplotlib offers a range of customization options to enhance your visualization, making them more informative and visually appealing.

New techniques learned:

Creating Advanced plot types: Beyond basic plots, matplotlib allows you to create a more advanced types of visualization such as pie charts, nose and heat maps, to represent data in different ways.

3) Enhancing and customizing plots: matplotlib provides various customization option to enhance the appearance and readability of your plots.

* Related Project/Practice work experienced and learned:

1) Practice work: 1) creating different plots, 1) line plot to visualize a simple data set, a bar chart to compare different categories, create a scatterplot to show the relation between two variables learned. Adding titles, labels, lines enhances the readability and inf.

* New Software/Machine/Tool/Equipment/Experiment learned:

1) Python interpreter: The core software needed to run and execute python code, download from python website 2) Integrated Development Environment to write and debug code efficiently like Jupyter notebook vs code pycharm

* Application of concept(s) (preferably real life scenario):

Matplotlib can be extremely useful: analysis, visualisation sells data for a retail store, at how to create line plots, bar charts, or plots to gain insight from the data

* Case Studies/Examples:

Monthly sales Trend

```
import matplotlib.pyplot as plt  
months = ['Jan', 'Feb', 'Mar']  
sales = [50, 55, 53]
```

```
plt.plot(months, sales)
```

```
plt.title("monthly sales")
```

```
plt.xlabel("months")
```

```
plt.ylabel("Sales in ₹")
```

```
plt.show
```

Assessment:

Marks Obtained: / 10

Signature of the Faculty:

Signature of the Student: Abhi

Name: K. Abhinav

Regn. No.: 241814100006



School: SO&T Campus: Villiana Gardens
Academic Year: 2015-16 Subject Name: D.A.U.P. Subject Code: 1013
Semester: 1st Program: BTech Branch: BCA Specialization: ...

Date:

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Creating Line, Bar and Scatter Plots
Learning Outcome: with mat plot lib.

Concepts learned (Mention 2/3 principles):

- 1) Understanding: Basic plot types: Different plot types serve different purposes in data visualization knowing when and how to use line plots, bar charts and scatter plots.
- 2) Customizing and enhancing plots: customizing plots by adding titles, labels, legends and using different styles enhance the readability and visual appeal of the data.

New techniques learned:

- 1) Line plot: Best for visualizing trends over time or continuous data
 - 2) Bar chart: ideal for comparing different
 - 3) Scatter plot: Great for showing relationships b/w two variables
- 2) Customizing and Enhancing plots: customization enhances the readability and visual appeal of plots. This includes adding titles, labels providing context and clarity to the visualization.

* Related Project/Practice work experienced and learned:

- 1) Practice work : 1) Creating a line plot: visualizing temperature over a year 2) Creating a Bar chart sales figure across different product categories.
- 3) Analyze the relation b/w hours studied and marks obtained.
- 2) Learned: customizing and enhancing plots by adding titles, labels to provide context and clarity using different b/w hours and marks.

* New Software/Machine/Tool/Equipment/Experiment learned:

Python interpreter: The core software needed to write, edit and execute code. Download from Python website. Integrated development environment: It is used to write, edit and debug code efficiently. Use Jupyter notebooks code.

* Application of concept(s) (preferably real life scenario):

Matplotlib can be extremely useful: Analyzing and visualising sales data for a store. Let's look at how to create line plots, bar charts and scatter plots to gain insights from the data.

* Case Studies/Examples:

```
monthly sales turned into a line plot
import matplotlib.pyplot as plt
Months = ['Jan', 'Feb', 'Mar']
sales = [50, 155, 155]
plt.title("Monthly sales")
plt.xlabel("Months")
plt.ylabel("Sales in ₹")
plt.show()
```

Assessment:

Marks Obtained: 9 / 10

Signature of the Faculty:

Signature of the Student: Abhi

Name: K. Abhinav

Regn. No.: 24131410006

* As applicable according
One sheet per topic



School:	S O E T	Campus:	Muthurajahamm
Academic Year:	2024-25	Subject Name:	D AWP
Semester:	1 st	Program:	B.TECH
Branch:	BCA	Specialization:
Date:		
Classroom Learning (Learning by Listening and Observations)			

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: customizing plots.

Learning Outcome:

Concepts learned (Mention 2/3 principles):

Basic customization: Basic customization involves adding and adjusting elements like titles, labels, legends and colours to enhance the readability and context of your plots.

Advanced customization: Advanced customization involves modifying the appearance of plots by changing line styles, markers, colours and adding annotations or guide to make the plots more effective.

New techniques learned:

1) Adding titles and labels provide essential information about what the plot represents making it easier for reviews 2) changing the line styles, colours and markers Different style make plots more visually appealing and help distinguish between different data series

* Related Project/Practice work experienced and learned:

Practice work, 1) create a line plot and add and labels to the axes. Create a plot with lines and add a legend to distinguish between them 3) customize the line style, colors and markers in a plot 2) learned Annotation and Grids annotations highly important

Classmate

* New Software/Machine/Tool/Equipment/Experiment learned:

1) Python interpreter: It is a software needed to run and execute python code. Download from the Python website 2) Integrated Development Environment to write, edit and debug code efficiently like Jupyter notebook vs PyCharm.

* Application of concept(s) (preferably real life scenario):

Matplotlib can be extremely useful for visualizing sales data for a retail store. We will look how to create line plots, bar charts and scatter plots to gain insights from the data.

* Case Studies/Examples:

```
Monthly sales trend  
pyplot  
import matplotlib.pyplot  
months = ['Apr', 'May', 'June']  
sales = [55, 57, 58]
```

```
plt.plot(months, sales)  
plt.title('monthly sales')  
plt.xlabel("months")  
plt.ylabel("Sales in $")  
plt.show()
```

Assessment:

Marks Obtained: 9 / 10

Signature of the Faculty:

Signature of the Student: Abhi

Name: K. Abhinav

Regn. No.: 20181100066

* As applicable across
One sheet per topic



School: SoE Campus: Villianuraram

Academic Year: 2021-22 Subject Name: DAVP Subject Code: 1013

Semester: 1st Program: BTech Branch: BCA Specialization:

Date: _____

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Plotting Data with Pandas and Matplotlib integration.
Learning Outcome:

Concepts learned (Mention 2/3 principles):

1) Leveraging Pandas for data handling: Pandas is a powerful data manipulation library that allows you to easily read, manipulate and clean data before plotting with matplotlib.

2) Plotting with matplotlib: It provides extensive capabilities for creating a wide range of plots combined with Pandas. It streamlines the process of visualizing data.

Techniques learned:

1) Plotting directly from pandas: Using pandas' built-in plotting capabilities with matplotlib's backend simplifies the creation of plots.

2) Customizing plots: Enhancing plots with titles, labels, legends, and custom styles improves readability and presentation.

* Related Project/Practice work experienced and learned:

- 1) Practice work: 1) Reading and inspecting data from a CSV file and inspect it.
2) manipulating Data: manipulate the data to calculate cumulative sales.
- 2) Learned: Data handling with Pandas and inspecting data using Pandas to inspect data functions.

* New Software/Machine/Tool/Equipment/Experiment learned:

Python interpreter: The core software needed to write and execute Python code. Download from the official Python website. 2) Integrated Environments: to write, edit and debug efficiently like Jupyter notebook, VS Code.

* Application of concept(s) (preferably real life scenario):

We can integrate Pandas and Matplotlib to visualize data. Imagine you're working on a sales report. You need to analyze monthly sales data to present the monthly to management. Sales Data Analysis and visualization.

* Case Studies/Examples:

Sales Data Analysis and visualization
`df = pd.read_csv('sales_data.csv')`
Output - month Sales

0	Jan	200
1	Feb	220
2	March	230
3	April	250

Assessment:

Marks Obtained: / 10

Signature of the Student: Abhi

Name: K. Abhinav

Regn. No.: 2413141000

Signature of the Faculty:

* As applicable across
One sheet per topic



School: SOC-T Campus: VIT Nagapattinam

Academic Year: 14-15 Subject Name: DAVL Subject Code: 1018

Semester: 1st Program: B.TECH Branch: BCA Specialization: _____

Date: _____

Classroom Learning

(Learning by Listening and Observations)

Note: Learning outcome will be measured through gain in knowledge, skill & attitude. Knowledge gain will be indicated through an answer to the question such as "why, when and where (application)". Skill gain will be reflected through an answer to the question "how to do things". Attitude change will be observed through visible alteration in behavior.

Name of the Topic: Introduction to plotly interactive visualization.
Learning Outcome:

Concepts learned (Mention 2/3 principles):

1) Creating Basic Interactive Plots: Plotly is a powerful library that allows for the creation of interactive plots that can be easily embedded in web application or shared online. 2) Enhancing and customizing to enhance the interactivity and visual appeal of plots. This includes Adding annotations, customizing hover information and adjusting the layout and styling of plots.

New techniques learned:

1) Setting up and Importing Plotly: 1) First you need to install Plotly. 2) Create an interactive line plot with Plotly.

2) Creating Basic Plots: 1) Create an interactive line plot with Plotly. 2) Create an interactive bar chart with Plotly.

3) Providing custom hover information makes plots more informative and user friendly.

* Related Project/Practice work experienced and learned:

1) Practicework: Setting up and creating interactive plots 1) Create a simple line plot and bar chartting 2) Enhancing customizing Interactive plots customizing line plot by adding hover information annotations and customizing the layout

* New Software/Machine/Tool/Equipment/Experiment learned:

1) Python interpreter: The core software to write and execute python code download from the official Python website
Development environment: To write and debug code efficiently like Jupyter notebook

* Application of concept(s) (preferably real life scenario):

Plotly can be used to create interactive visualization. Imagine you are tracking your daily step count and want to visualize to use your progress over a week. This example will demonstrate how to use Plotly to create an interactive Bar chart.

* Case Studies/Examples:

```
Import plotly.graph_objs as go
Import plotly.offline as py
Days = ["Monday", "Tuesday", "Wednesday"]
steps = [3000, 3200, 2500]
Trace = go.Bar(x=Days, y=steps)
Data = [Trace]
Layout = go.Layout()
Fig = go.Figure(data=Data, layout=Layout)
py.iplot(Fig, filename="BarChart.html")
```

Assessment:

Marks Obtained: 9/10

Signature of the Faculty:

Signature of the Student: 

Name: 10-Abhinav

Regn. No.: 20131410006

*An application form
One sheet per page

ASSESSMENT

Classroom Learning

Topic	Full Mark	Marks Obtained
pic - 1	10	
pic - 2	10	
pic - 3	10	
pic - 4	10	
pic - 5	10	
pic - 6	10	
pic - 7	10	
pic - 8	10	
pic - 9	10	
pic - 10	10	
pic - 11	10	

Topic	Full Mark	Marks Obtained
Topic - 12	10	
Topic - 13	10	
Topic - 14	10	
Topic - 15	10	
Topic - 16	10	
Topic - 17	10	
Topic - 18	10	
Topic - 19	10	
Topic - 20	10	
Average Total	10	

Signature of the Faculty

COMPONENT-WISE LEARNING OUTCOMES (Continuous Assessment): (Descriptive)

Learning from Record of Classroom Learning:

Learning from Assignment:

Learning from Presentation:

Learning from Mid Sem Exam:

Abhinav.
Signature of the Student

Page No.....
*One sheet per learning record to be used

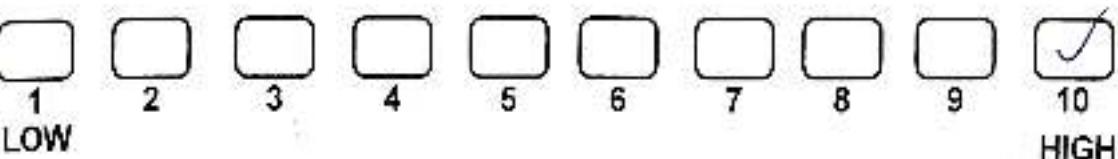
COURSE OUTCOMES (COs) ATTAINMENT**Expected Course Outcomes (COs):**

(Refer to COs Statement in the Syllabus)

- (CO) able to gain knowledge
 visualization with good
 story line and performance of
 data analyst

Course Outcomes (COs) Attained:

How would you rate your learning of the subject based on the specified COs?

**Learning Gap (If any):**

No learning gap

Books/Manuals Referred:

Te: _____

K. Abhinav.
Signature of the Student**Suggestions / Recommendations:**

(by the Course Faculty)

Signature of the Faculty

Page No.

*One sheet per learning record to be used



Centurion
UNIVERSITY

*Simplicity. Discernment.
Innovative. Competitive.*

**CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT
ANDHRA PRADESH**



VIT UNIVERSITY COLLEGE OF ENGINEERING AND TECHNOLOGY

Record of Applied and Action Learning (Programming Practice)

Subject Name: Python Programming

Subject Code: CUGC-1003

Semester: Ist

Name: K. Abhinav

Registration No.: 2013110006

Program/Branch: DIA

Specialization: _____

Academic Year: 2014-15

Campus: Vizianagaram



Centurion
UNIVERSITY

ENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT
ANDHRA PRADESH

Certificate

This is to certify that Mr./Ms. K. Akhila ... having
Registration No. 201121M100006 of 3rd Semester,
B.TECH - BCA Program,
School, Vizianagaram Campus has completed
number of experiments in Applied and Action
Learning Laboratory and fulfils the course
requirements.

Signature of the HoD/Dean

Signature of the Faculty

Office Seal

INDEX

Date	Name of the Experiment	Page No.	Remark	Faculty Signature
	write a Python program to declare and use variables different data types int float string			
	Create a Python to execute the demonstrate a format of arithmetic and logical operators			
	Implement conditional statement elif if to control program flow			
	Design and Python functions to perform specific tasks including parameters.			
	Develop programs to use nested loops and functions to solve the complex numbers.			
	Create plots and various charts using ^{lib} Python comparing different type of plots			
	Pseudo code for Pre Plot			
	Implementing area charts and piecharts (creating histogram bar plots)			
	Generating basic plots using Matplotlib			
	Representation of map bubble chart and heat			
	Pseudo code for reading CSV files			

School: SDET Campus: VZM
Academic Year: 2024-25 Subject Name: DAVP Subject Code: CVB6108
Semester: 1 Program: Btech-BCA Branch: BCA Specialization: BCA

Date: _____

Applied and Action Learning

(Learning by Doing and Discovery)

of the Experiment: write python program to declare and use variables
of different data types (int,float,string) and
Using Phase: Pseudo Code / Flow Chart / Algorithm basic operators.

art

input a,b,c i.e two numbers from the user

then use basic operations of arithmetic
using the two variables

sum, difference, product, quotient, power,
and floor division

print the code and get output

END.

* Testing Phase: Compilation of Code (error detection)

a = int(input ("Enter a number"))

b = int(input ("Enter a number"))

Arithmetic operators

Addition

print ("sum of a and b : " a+b)

subtraction

print ("difference of a and b : " a-b)

Multiplication

print ("product of a and b : " a*b)

Division

print ("quotient of a and b : " a/b)

floor-multiplication

print ("Module of a and b : " a%b)

floor-division

print ("floor division of a and b : " a//b)

Implementation Phase: Final Output (no error)

Enter a number : 5

Enter a number : 6

sum of a and b : 11

Difference of a and b : -1

Product of a and b : -1

Quotient of a and b : -0.8333333333333334

Module of 'a' and 'b' : 15.625

Floor division of a by b : 0

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Conceptual Understanding	10		
Planning and Execution/ Technical Simulation/ Programming	10		
Result and Interpretation	10		
Overall Application of Applied and Action Learning	10		
	10		
Total	50		

Signature of the Student: *A. Abhishek*

Name: I.C.A. Abhishek

*As applicable according to the experiment.

Two sheets per experiment Page No. 1/2 used.

Regn. No.: 111111100006

School: D.E.I Campus: VIT
Academic Year: 2019-20 Subject Name: DPVP Subject Code: CSE-1006
Semester: 1 Program: Btech BBA Branch: BBA Specialization: ICT

Date: _____

Applied and Action Learning

(Learning by Doing and Discovery)

the Experiment:

Working Phase: Pseudo Code / Flow Chart / Algorithm

= float

for t

Assign values to variables x and y

Perform arithmetic operations

Apply expressions

Print the results

end.

string

start

Assign a string to a variable

Access characters by index

Print the last character of w and

enter string.

Print a sub string of w from index i to j
in a string

String Concatenation by adding two
strings with addition operation

Print result

end.

* Testing Phase: Compilation of Code (error detection)

x = 3 + 4

y = 2 * 3

Arithmetic operation

a = x + y

b = x - y

c = x * y

d = x / y

print("sum: a")

print("Difference b")

print ("Product c")

print ("Quotient d")

w = "Time management"

Access characters by index

print(w[4])

print(w[5:11])

String

print(w[7:12])

String concatenation

a = "Hi"

b = "Alice"

c = a + b

print(c)

Implementation Phase: Final Output (no error)

Sum : 5 x 5

Difference : 0.43

Product : 8.50

Quotient : 1.16

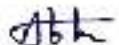
Time management

Management

Hi Alice

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Implementation	10		
Planning and Execution/ Actual Simulation/ Programming	10		
Analysis and Interpretation	10		
Overall Application of Applied and Action Learning	10		
	10		
	50		

Signature of the Student: 

Name: Abhi Ray *As applicable according to the experiment.

Regn. No.: Two sheets per experiment Pg no. / No. used.

<41814100006

School: 8067

Campus: K2M

Academic Year: 2024-25 Subject Name: DAVP Subject Code: CPM0102

Semester: 1 Program: Batch-BCA Branch: BCA Specialization: BCA

Date:

Applied and Action Learning

(Learning by Doing and Discovery)

the Experiment: create a python to execute demonstrate the use
of various Operators arithmetic relation logic
Phase: Pseudo Code / Flow Chart / Algorithm

arithmetic operators

start

set num1=10 num2=5

set addition =num1+num2

set subtraction =num1-num2

set multiplication = num1 * num2

set division = num1 / num2

set floor - division = num1//num2

set modulo = num1 % num2

set exponentiation = num1 ** num2

print all the values accordingly.

end.

* Testing Phase: Compilation of Code (error detection)

1. Arithmetic operators

num1 = 10

num2 = 5

2. Arithmetic operations

Addition = num1 + num2

Subtraction = num1 - num2

Multiplication = num1 * num2

Division = num1 / num2

Floor division = num1 // num2

Module num1 % num2

Exponentiation = num1 ** num2

print ("Addition = " addition)

print ("Subtraction = " subtraction)

print ("Multiplication = ", multiplication)

print ("Division : division")

print ("Floor division: floor-division")

print ("Module ~ Module")

print ("Exponentiation = " exponentiation)

Implementation Phase: Final Output (no error)

Addition = 15

Subtraction = 5

Multiplication = 50

Division = 20

Floor division = ~

Module = 0

Exponentiation = 100000

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Design and Execution/	10		
Final Simulation/ Programming	10		
Analysis and Interpretation	10		
Overall Application of Applied and Action Learning	10		
	10		
	50		

Signature of the Student: Abhi

Name: Ic. Abhinav.

*As applicable according to the experiment.

Two sheets per experiment. Page No. / Note used.

Regn. No.: 20161410006

* Testing Phase: Compilation of Code (error detection)

a = 33

b = 66

H Relation operations

lc = a == b

l = a != b

m = a > b

n = a < b

o = a >= b

p = a <= b

print ("equal to : " lc)

print ("Not equal to : " l)

print ("less than : " n)

print ("greater than or equal to : " o)

print ("less than or equal to : " p)

a = true

b = false

H Logical operator

x = a and b

y = a or b

not a

print(x)

print(y)

print(z)

Implementation Phase: Final Output (no error)

equal to false

not equal to true

greater than false

less than true

Greater than or equal to false

less than or equal to true

false

true

false

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Designing and Execution/ Numerical Simulation/ Programming	10		
and Interpretation	10		
Usage of Applied and Action Learning	10		
	10		
	50		

Signature of the Student: Abhi

Name : K. Abhirav

*As applicable according to the experiment.

Two sheets per experiment. Page No. _____ used.

Regn. No. : 241014100006

School: SMET Campus: VZN

Academic Year: 2014-15 Subject Name: DAVP Subject Code: 108L-1000

Semester: 1 Program: B.Tech-BCA Branch: BCA Specialization: BCA

Date: _____

Applied and Action Learning (Learning by Doing and Discovery)

the Experiment:

Working Phase: Pseudo Code / Flow Chart / Algorithm

integer to String

Start

Set a=42

Set b= str(a)

Print b

End

float to string

Start

Set x=3.14

Set y = string representation of x

Print y

End.

* Testing Phase: Compilation of Code (error detection)

a = 42

b = str(a)

print(b)

x = 3.14

y = str(x)

print(y)

Implementation Phase: Final Output (no error)

123

214

451

316

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Designing and Execution/ Numerical Simulation/ Programming	10		
Result and Interpretation	10		
Overall Applied and Action Learning	10		
	10		
	50		

of the Faculty

Signature of the Student: AbhishekName: A. AbhishekRegn. No.: 201814160006*As applicable according to the experiment.
Two sheets per experiment. Please勿use.

School: SEZ

Campus: VZN

Academic Year: 2024-25 Subject Name: DIVP Subject Code: CISC-1005

Semester: I Program: B.tech BCA Branch: BCA Specialization: BCA

Date: _____

Applied and Action Learning

(Learning by Doing and Discovery)

the Experiment: Implement Conditional Statement if else if else
for while) in Python to Control program flow.

Phase: Pseudo Code / Flow Chart / Algorithm

conditional statements

start

if num is greater than 0 then print "Positive number"

else if num is less than 0 then print "Negative number"

else print "Zero"

end of

end

for loop

start

for i=0 to 4 print i

end for

end

while loop

start

set i=0

while i is less than 5 print i

* Testing Phase: Compilation of Code (error detection)

3) End while

```
num = int(input("Enter a no."))
if num > 0:
    print ("Positive number")
elif num < 0:
    print ("Negative number")
else:
    print ("zero")
for i in range(5):
    print(i)
i = 0
while i < 5:
    print(i)
    i += 1
```

Implementation Phase: Final Output (no error)

Enter a no.: 10
positive number

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Design and Execution/	10		
Simulation/ Programming	10		
and Interpretation	10		
of Applied and Action Learning	10		
	10		
	50		

of the Faculty

Signature of the Student: Abhi

Name: K. Abhilash
*As applicable according to the experiment.
 Regn. No.: 2018111111
Two sheets per experiment. Pupils/Neobusued.

School: SWT Campus: VZN
Academic Year: 2021-22 Subject Name: DAVP Subject Code: CVAC1008
Semester: I Program: BCA Branch: BCA Specialization: BCA

Date:

Applied and Action Learning

(Learning by Doing and Discovery)

The Experiment: Design and Python functions to perform
specify tasks including parameters return
Working Phase: Pseudo Code / Flow Chart / Algorithm

dd-num

function add-num (num1, num2)

Set sum with + operator

Print 'sum' sum.

End function

nd-square

function find-square (num)

Set result num (num)

Return result

End function

Set square = find-square (3)

Print "Square": square

* Testing Phase: Compilation of Code (error detection)

```
def add_num(num1, num2)
    sum = num1 + num2
    print("sum : " + str(sum))
add_num(5, 4)
```

```
def find_square(num)
    result = num * num
    return result
square = find_square(3)
print("square : " + str(square))
```

Implementation Phase: Final Output (no error)

Sum : 9

Date : 09

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Designing and Execution/	10		
Model Simulation/ Programming	10		
Analysis and Interpretation	10		
Overall Application of Applied and Action Learning	10		
	10		
	50		

Signature of the Student: *Abs*Name: *Ic. Abhinav*

*As applicable according to the experiment.

Two sheets per experiment. *No 20/Rehearsed*Regn. No.: *141814160006*

School: 8067 Campus: VZN
Academic Year: 2024-25 Subject Name: DAVP Subject Code: C18C1008
Semester: I Program: Btech-BIA Branch: BCA Specialization: BCA

Date: _____

Applied and Action Learning

(Learning by Doing and Discovery)

The Experiment: Develop program to use nested loops and functions to solve complex problems.
Working Phase: Pseudo Code / Flow Chart / Algorithm

function is_prime (num)

 num is less than or equal to 1 then

 return false

 end if

 if num is divisible by then return

 false

 And for return true

 end function

start

for num = 1 to 20

 if is_prime(num) is true then print

 if is_prime(num) is true then print
 num is a prime no

 end if end for

end

* Testing Phase: Compilation of Code (error detection)

```
def is_prime
```

```
def is_prime
```

```
    if num <= 1
```

```
        for i in range
```

```
            if num % i == 0
```

```
                return False
```

```
        return True
```

* Example

```
for num in range(1, 21)
```

```
    if is_prime(num)
```

```
        print(f'{num} is a prime')
```

Implementation Phase: Final Output (no error)

s a prime no
 is a prime no

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Design and Execution/	10		
Final Simulation/ Programming	10		
Analysis and Interpretation	10		
Overall Application of Applied and Action Learning	10		
	10		
	50		

Signature of the Student: Abhi

Name: ICI Abhinav
 Regn. No.: 241814100066
*As applicable according to the experiment.
 Two sheets per experiment. Page No. 1 of 2.

School: SOE

Campus: M.I.T. Nanogaram

Academic Year: 10/14-15, Subject Name: RAUP, Subject Code: 1003

Semester: Ist, Program: BTech, Branch: BCA, Specialization:

Date:

Applied and Action Learning

(Learning by Doing and Discovery)

of the Experiment: creating plots and various charts using
Python libraries

Phase: Pseudo Code / Flow Chart / Algorithm

~~scatter plot pseudo code:~~

~~start~~

~~import matplotlib~~

~~define x and y data~~

~~create scatter plot with
color and marker~~

~~set title and labels~~

~~Show plot~~

~~End~~

~~histogram pseudo code~~

~~start~~

~~import matplotlib~~

~~definedata list~~

~~create histogram~~

~~with color and edge colour~~

~~set title as "hist plot"~~

~~label axes~~

~~Show plot~~

~~End~~

* Line chart

Pseudo code:

1) Start

2) Import matplotlib

3) Define x and y

4) Create a line plot
with data points

5) Label x-axis and
y-axis

6) Add line plot

title

7) Display the plot

8) End

* Pseudo code for barcharts.

1) Start

2) Import matplotlib

3) Define list categories

4) Define values

5) Create bar plot
with x-axis and y-axis

6) Set the title

7) Label x-axis and y-axis

8) Display the plot

9) End.

* Testing Phase: Compilation of Code (error detection)

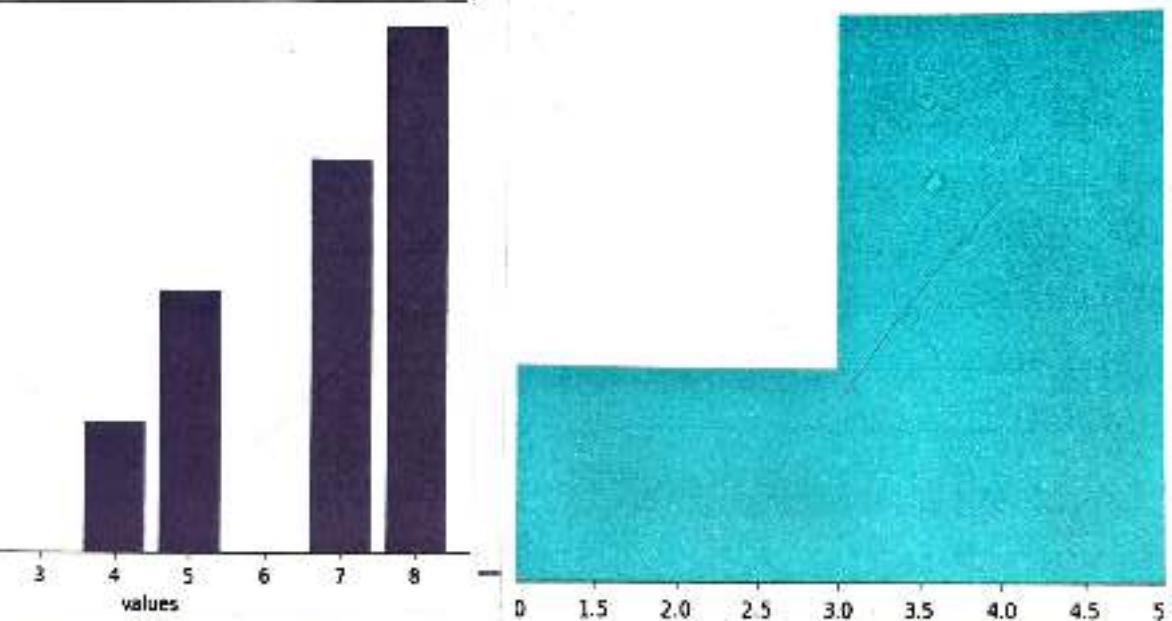
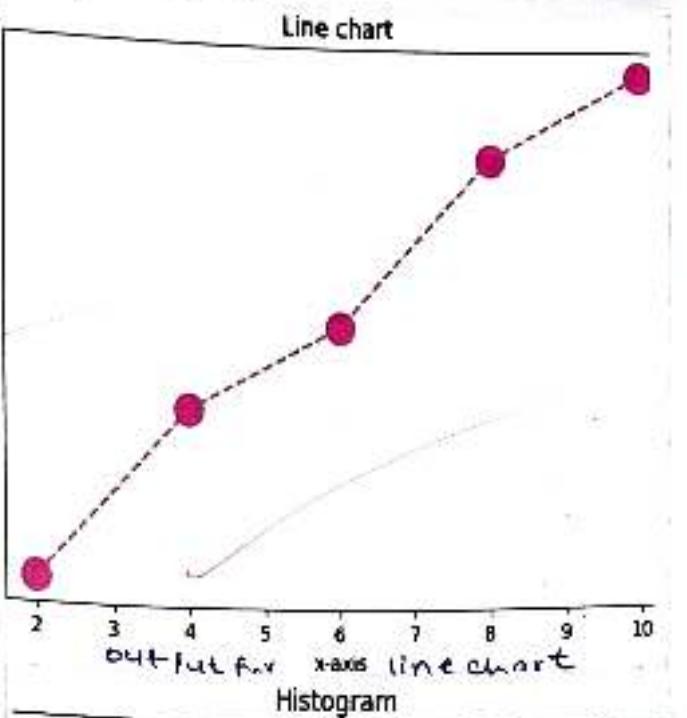
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
a = [15, 30, 45, 60, 75]
b = [0, 13, 24, 45, 46]
plt.scatter(a, b, color='lightcoral')
plt.title('Scatter Plot')
plt.show()

# Bar Charts
categories = ['A', 'B', 'C', 'D']
values = [1, 4, 5, 7]
plt.bar(categories, values, color='pink')
plt.title("Bar chart")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()

# Line Chart
x = [0, 4, 6, 8, 10]
y = [1, 5, 7, 11, 13]
plt.plot(x, y, markers=True, sizes=15)
color = 'magenta'
line
style '--'
marker = 0
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.title('Linechart')
plt.show()

# Histogram
data = [1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5]
plt.hist(data, color='turquoise')
plt.title("Histogram")
plt.show()
```

Implementation Phase: Final Output (no error)



Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Design and Execution/	10		
Final Simulation/ Programming	10		
Analysis and Interpretation	10		
Overall Application of Applied and Action Learning	10		
	10		
	50		

of the Faculty

Signature of the Student: *A. Abhinav*Name: *A. Abhinav*

As applicable according to the experiment.

Two sheets per experiment. Please use both sides.

Reg. No.: *2418141000076*

School: SAET

Campus: Vijayanagaram

Academic Year: 2024-25 Subject Name: DAUP Subject Code: 1002

Semester: 1st Program: B.TECH Branch: BCA Specialization: _____

Date: _____

Applied and Action Learning

(Learning by Doing and Discovery)

the Experiment: Comparing different types of plots

Phase: Pseudo Code / Flow Chart / Algorithm

plot Pseudo code:

import matplotlib

define categories and values

define colors

create a bar chart with (colors edgercolors)

Add title

Add labels

Display the chart

scatter plot Pseudo code:

import matplotlib

define x-value

define y-value

Create a scatter plot with (x and y values, color marker)

Add title

Add labels

Display count

* Testing Phase: Compilation of Code (error detection)

```
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt
```

Bar plot:

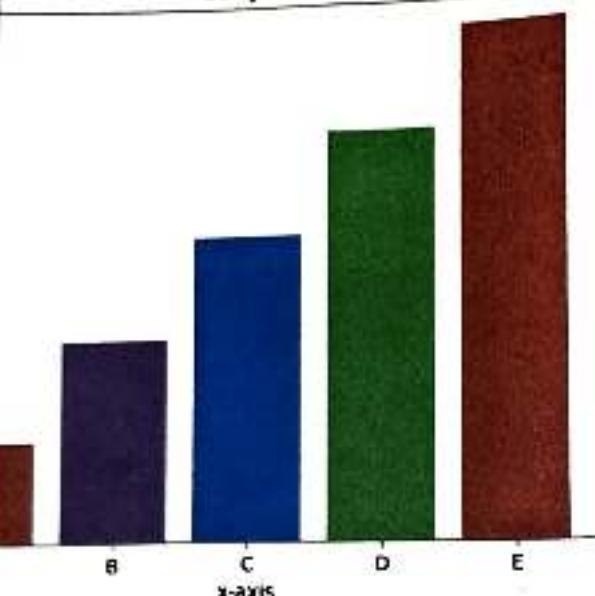
```
categories = ['A', 'B', 'C', 'D', 'E']  
values = [30, 60, 90, 120, 150]  
color = ['darkred', 'indigo', 'blue', 'green', 'yellow']  
plt.bar(categories, values, color=color)  
plt.title('Barplot')  
plt.xlabel('x-axis')  
plt.ylabel('y-axis')  
plt.show()
```

scatter plot

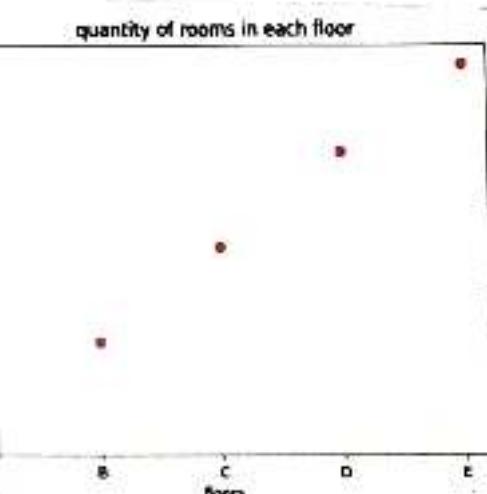
```
x = ["A", "B", "C", "D", "E"]  
y = [22, 44, 66, 88, 108]  
plt.scatter(x, y, color='red')  
plt.title('Quantity of rooms in each floor')  
plt.xlabel("Floors")  
plt.ylabel("Rooms")  
plt.show()
```

Implementation Phase: Final Output (no error)

bar plot



output for bar plot:



output for scatter plot:

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
pt	10		
ng and Execution/ cal Simulation/ Programming	10		
and Interpretation	10		
d of Applied and Action Learning	10		
	10		
	50		

of the Faculty

Signature of the Student: *Abhi*Name: *K. Abhinav*Regn. No.: *241814100006** As applicable according to the experiment.
Two sheets per experiment.

Programs used:

School: SOFT Campus: vizianagaram

Academic Year: 2024-25 Subject Name: DAvP Subject Code: 1008

Semester: 1st Program: BTech Branch: BCA Specialization:

Date:

Applied and Action Learning (Learning by Doing and Discovery)

the Experiment: Pseudo code for Pie Plot

Phase: Pseudo Code / Flow Chart / Algorithm

Pseudo code for Pie Plot:

Start

import matplotlib.pyplot

Define categories and values

Define explode and colors

Create a pie chart with (output, shadow, explode, colors)

Set title as Pie Plot

Show plot

end.

Pseudo code for Bar Plot:

Start

import matplotlib.pyplot

Define categories list

Define values list

Create bar plot with ((color, edge_color))

Set title as "BarPlot"

Label the axes

Show plot

end.

* Testing Phase: Compilation of Code (error detection)

```
import numpy as np
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

Bar Plot:

```
categories = ['AIML', 'CIC', 'CSD', 'B(A)', 'B(C)']  
values = [100, 200, 300, 400, 500]
```

```
plt.bar(categories, values, color = 'blue')  
plt.title('barplot')  
plt.xlabel('domains')  
plt.ylabel('values')  
plt.show()
```

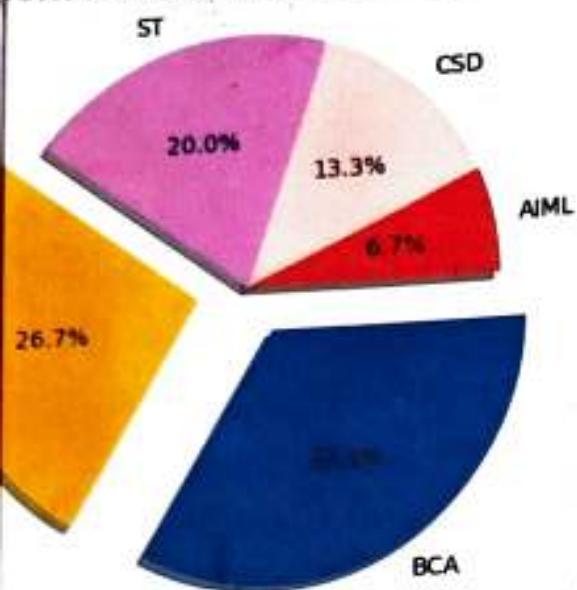
Pie Plot:

```
categories = ['AIML', 'CSD', 'ST', 'CN', 'B(A)', 'B(C)']  
values = [100, 200, 300, 400, 500]  
explode = [0, 0, 0, 0.2, 0.2]
```

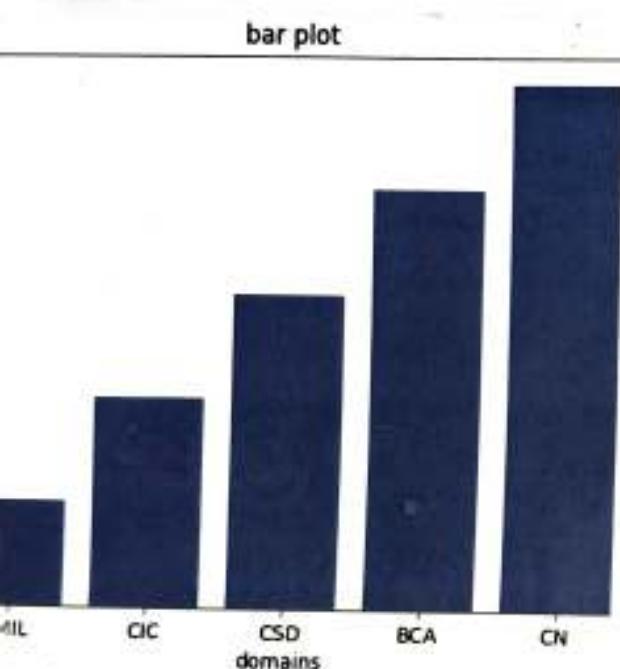
```
colors = ['red', 'pink', 'violet', 'orange', 'blue', 'green']  
plt.pie(values, labels=categories, autopct='%.2f%%',  
        shadow=True, explode=explode)
```

```
plt.show()
```

Implementation Phase: Final Output (no error)



out put for pieplot
showing categories
of domains



out put for bar
plot showing
domains.

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Plot	10		
Designing and Execution/ Virtual Simulation/ Programming	10		
Analysis and Interpretation	10		
Total of Applied and Action Learning	10		
	10		
	50		

Signature of the Student: *Abha*

Name: *Ie. Abhinav*

*As applicable according to the experiment.
Two sheets per experiment. Page No. _____

Regn. No.: *201814106666*

School: SDET Campus: Vizianagaram
Academic Year: 2024-25 Subject Name: DAUP Subject Code: 1003
Semester: IST Program: B.TECH Branch: BCA Specialization:

Date: **Applied and Action Learning**
(Learning by Doing and Discovery)

the Experiment: Implementing area plots pie charts
Working Phase: Pseudo Code / Flow Chart / Algorithm

area plot :

Import mat plot lib library.

Create lists of x1,y1,y2 for the data

Create first area plot between x,y with
colour and transparency.

Create second area plot between x1,y2
by colour

Set title for plot

Label x-axis and y-axis

Add a legend to differentiate b/w two areas.

Display the plot.

chart :

Import mat plot lib library

Create a list for labels sizes colors

Create a pie chart for (sizes, explode, labels,
colours, percentage shadow)

Set the title for the plot

Display the plot.

* Testing Phase: Compilation of Code (error detection)

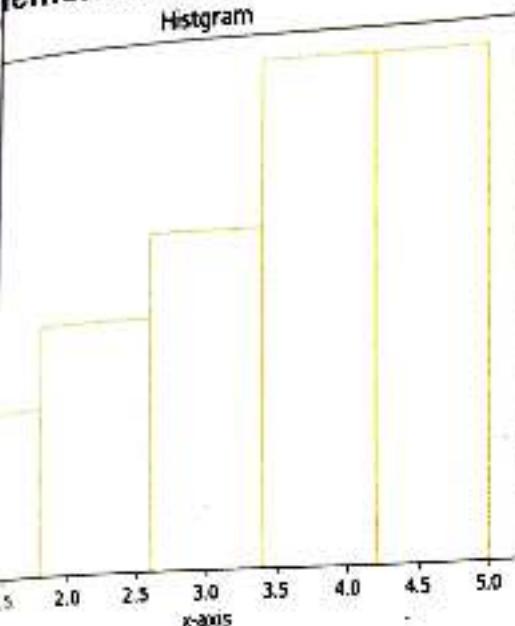
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt.

# histogram
data = [1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5]
plt.hist(data, color='white', edgecolor='or',
          bins=5)
plt.title ("Histogram")
plt.xlabel ("x-axis")
plt.show()

# BarPlot
categories = ['APPLE', 'SAMSUNG', 'OPPO', 'INFINITI']
values = [15, 80, 35, 30, 25]
plt.bar(categories, values, colour='grey')
plt.xlabel ("domain")
plt.ylabel ("values")
plt.show()
```

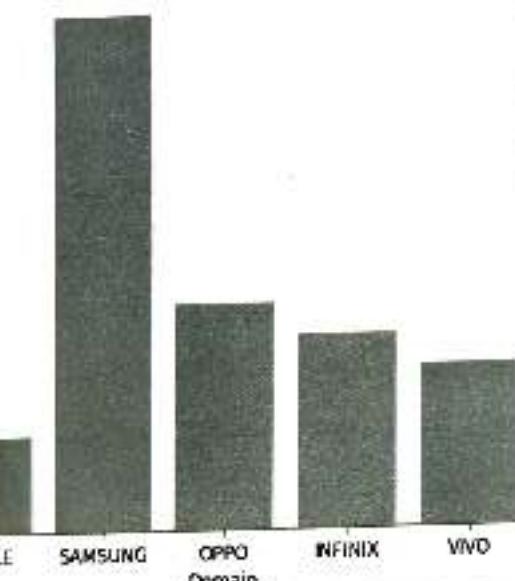
Implementation Phase: Final Output (no error)

Histogram



out put for hist
plot.

bar plot



out put for bar
graph of mobile
brands

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Designing and Execution/	10		
Actual Simulation/ Programming	10		
Analysis and Interpretation	10		
Evaluation of Applied and Action Learning	10		
	10		
	50		

Signature of the Student: *Akhil*

Name: *Le. Akhil Kumar*

*As applicable according to the experiment.

Two sheets per experiment. Page 1/2 to be used.

Regn. No.: *201914106666*

School: SOET Campus: Vizianagaram

Academic Year: 2024-25 Subject Name: DAVP Subject Code: _____

Semester: 1st Program: BTech Branch: BCA Specialization: 100%

Date: _____

Applied and Action Learning

(Learning by Doing and Discovery)

The Experiment: Generating basic plots using matplotlib

Phase: Pseudo Code / Flow Chart / Algorithm library

Pseudo code for line Plot

import mat plot lib

fine data list

create a line plot

label x-axis and

y-axis

dd line plots title

isplay the Plot

;

Pseudo code for scatter plot

1) Start

2) import mat plot lib

3) Define xandy

4) Create a scatter plot
with colour and marker

5) Set tittle and labels

6) Show plot

7) end.

Pseudo code for bar Plot

import

port matplot lib

ne list of categories

ine values

te barplot withxandy 8) set tittle as his plot

the title

sel "x y" axis

isplay the plot

end.

Pseudo code for histogram

1) Start

2) import mat plot lib

3) define list of categories

4) Create histogram with
colour and edge colour

5) Set tittle as his plot

6) Label axes

7) Show plot

8) end.

* Testing Phase: Compilation of Code (error detection)

Applied and Actual

```
import numpy as np
import pandas as pd
import matplotlib as lib
    pyplot as plt
```

line plot:

```
a = [2, 4, 6, 8, 9]
b = [6, 13, 24, 45, 46]
```

```
plt.plot(a, b)
```

```
plt.xlabel("a-values")
```

```
plt.ylabel("b-values")
```

```
plt.title("Lineplots")
```

```
plt.show()
```

Barplot

```
categories = ["Aim1", "sd", "n", "sw", "BC"]
```

```
values = [1.5, 6, 5, 1.5, 3.5, 9]
```

```
plt.bar(categories, values,
        color="black")
```

```
plt.title("barplot")
```

```
plt.xlabel("Domain")
```

```
plt.ylabel("values")
```

```
plt.show()
```

Scatterplot

```
a = [2, 4, 6, 8, 9]
b = [6, 13, 24, 45, 46]
```

```
plt.scatter(a, b, color="blue")
```

= 'blue' marker

```
plt.title("scatter pl")
```

```
plt.show()
```

Histogram

```
data = [1, 1, 2, 2, 2,
        3, 4, 4, 4, 4, 5,
```

```
plt.hist(data,
         color="red")
```

```
color="black")
```

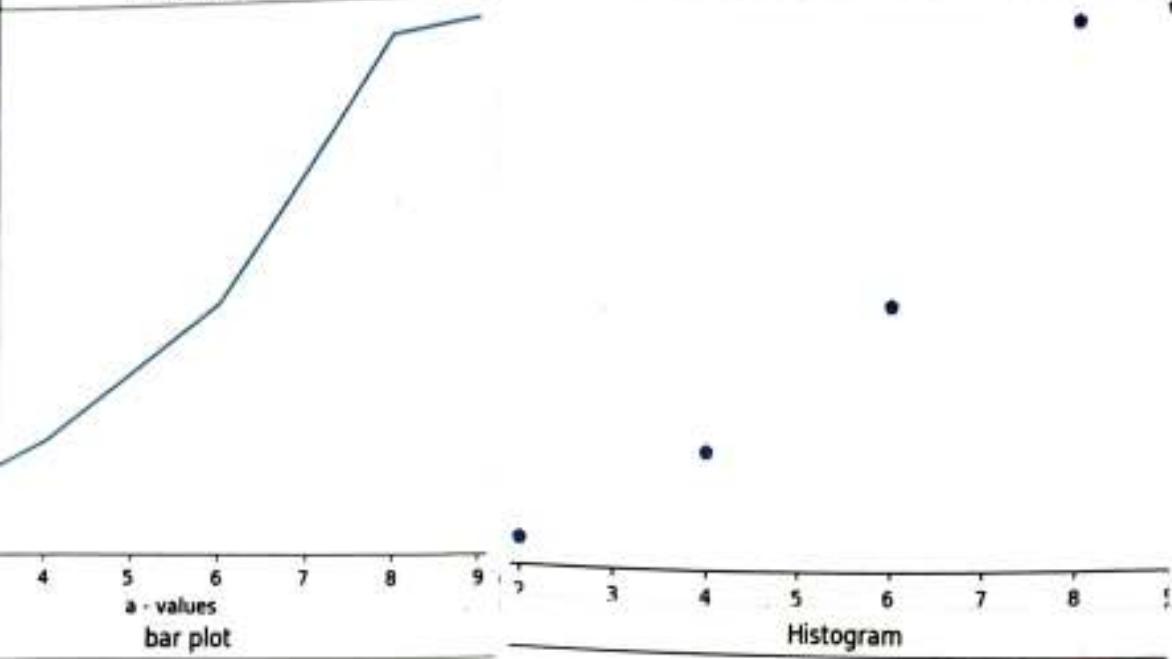
```
plt.title("Histo")
```

```
plt.xlabel("ax")
```

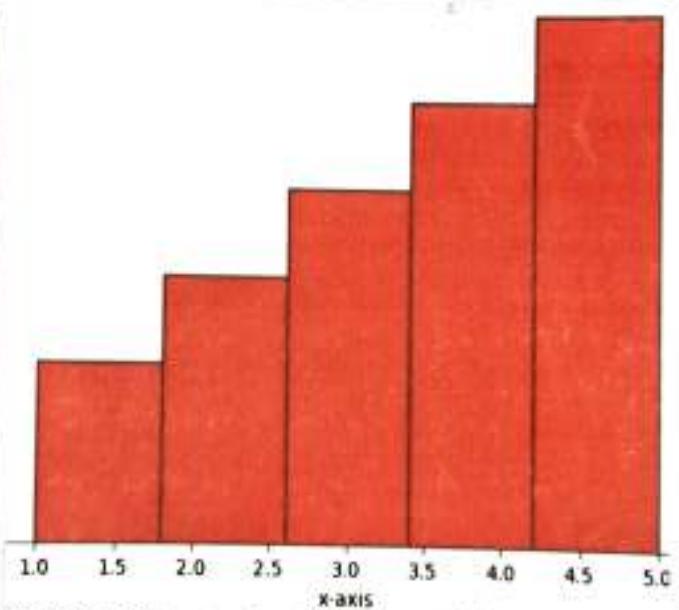
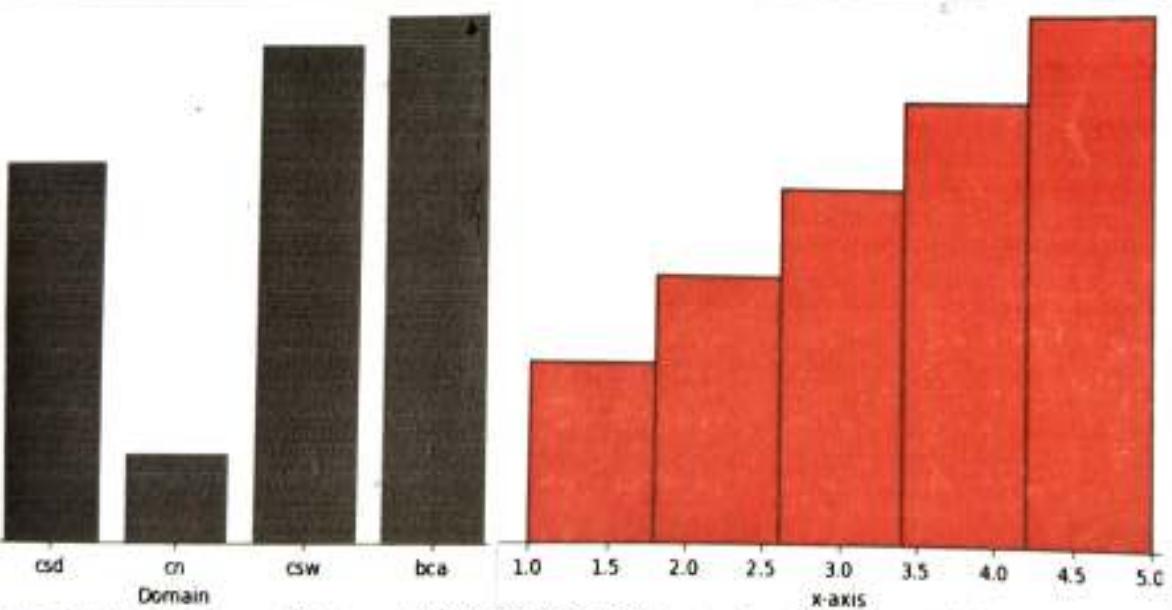
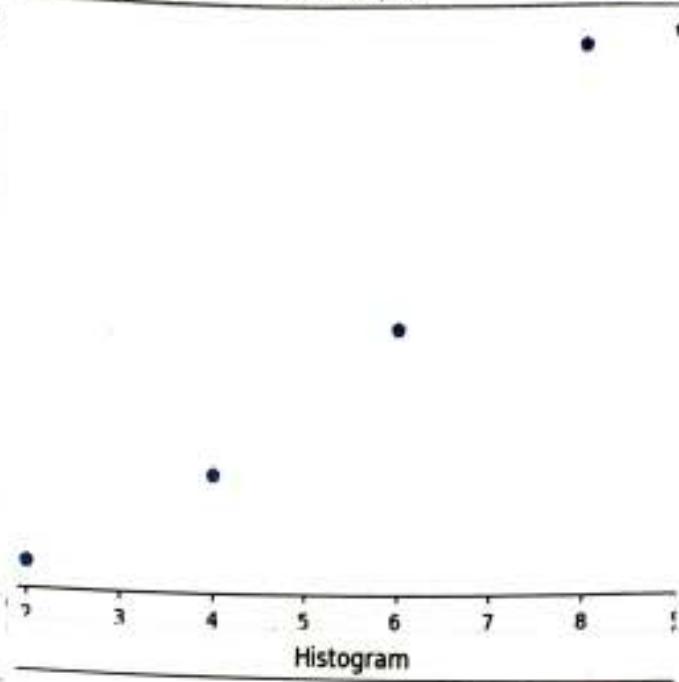
```
plt.show()
```

Implementation Phase: Final Output (no error)Output of line plot

Line plots

Output of scatter plot

Scatter plot

Output of barplotASSESSMENT output of histogram

Rubrics	Full Mark	Marks Obtained	Remarks
1	10		
g and Execution/	10		
al Simulation/ Programming			
and Interpretation	10		
of Applied and Action Learning	10		
	10		
	50		

Signature of the Student: Abhi

Name: K. Abhinav

*As applicable according to the experiment.
Two sheets per experiment. P10/20/Nebo used.

Regn. No.: 241314100006

School: SOET Campus: Vizianagaram

Academic Year: 2024-25 Subject Name: DAuf Subject Code: 1003

Semester: 1st Program: B.TECH Branch: BCA Specialization:

Date:

Applied and Action Learning

(Learning by Doing and Discovery)

The Experiment: Representation of bubble chart & heat map
Working Phase: Pseudo Code / Flow Chart / Algorithm

do code for bubble chart:

Import matplotlib.pyplot
Create lists x,y size for co-ordinates and bubbles
Plot bubble chart using the x,y and size lists
and specify and colour and transparency
Set the title of plot
Label x-axis and y-axis
Display the plot.

do code for heat map

Import the sea born matplotlib lib.
numpy lib

Create 10x10 matrix of random data using
numpy

Plot heat map with 'coolwarm' and 'color'

Set the title of plot

Label x-axis and y-axis

Show the plot.

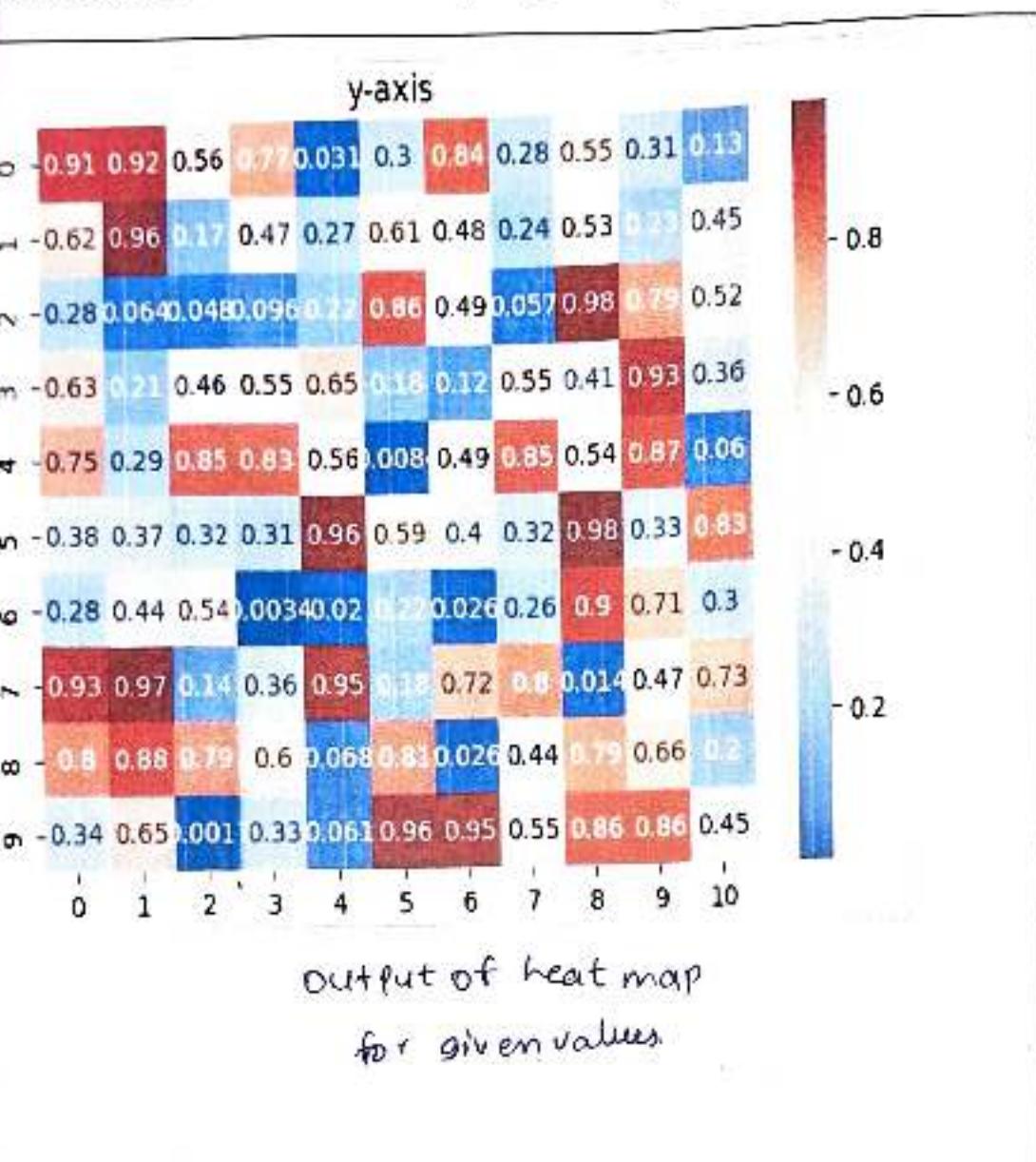
* Testing Phase: Compilation of Code (error detection)

Applied and basic

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [5, 3, 2, 1, 4]
size = [50, 90, 1100, 73, 56]
plt.scatter(x, y, s=size, color='orange', alpha=0.5)
plt.title('Bubble Plot')
plt.xlabel("x")
plt.ylabel("y")
plt.show()

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
data = np.random.rand(75, 39)
sns.heatmap(data, cmap='coolwarm')
plt.title("Heatmap")
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.show()
```

Implementation Phase: Final Output (no error)

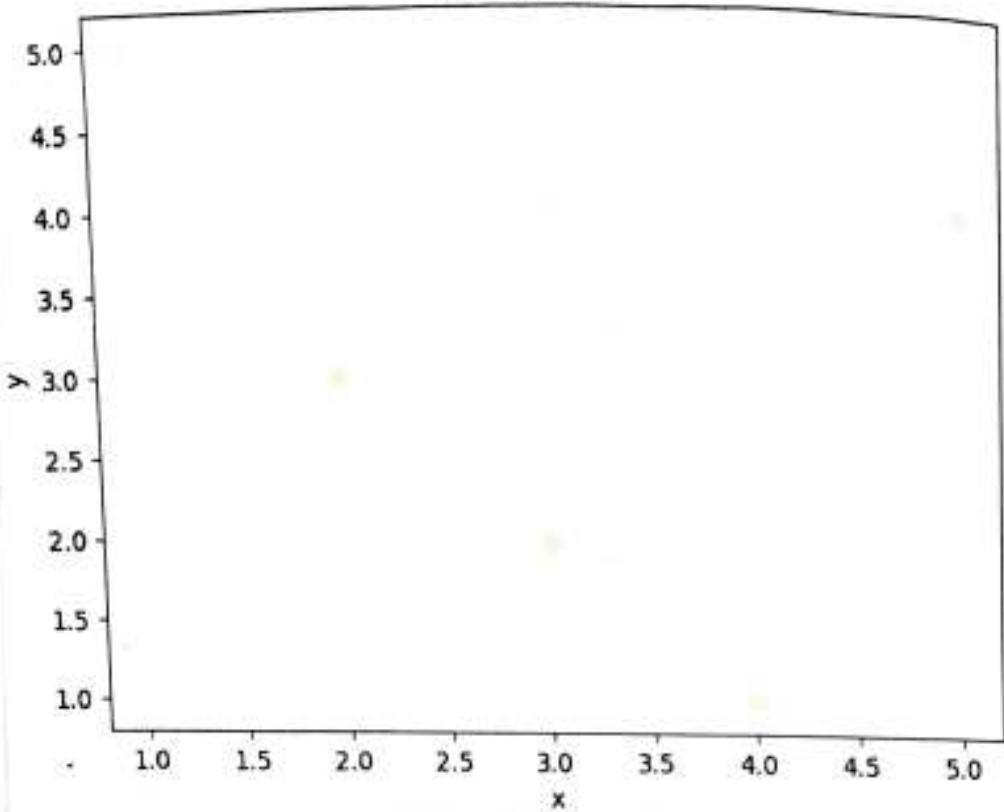


ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Report	10		
Designing and Execution/	10		
Technical Simulation/ Programming	10		
Result and Interpretation	10		
Total of Applied and Action Learning	10		
	10		
	50		

Signature of the Student: *A. Abhinav*Name: *20191410506* Ic. Abhinav* As applicable according to the experiment.
Two sheets per experiment. Page no. 1 used.Regd. No.: *241814100006*

Bubble Plot



Output of Bubble Plot

for given values.

School: SOET Campus: Vizianagaram

Academic Year: 2024-25 Subject Name: DAUP Subject Code: 1003

Semester: 1st Program: BTech Branch: BCA Specialization:

Date:

Applied and Action Learning

(Learning by Doing and Discovery)

(the Experiment: Pseudo code for reading csv files

ing Phase: Pseudo Code / Flow Chart / Algorithm

Pseudo code for reading csv files:

import Pandas library

Read the csv file into a data frame using
pd.read_csv.

print the data frame

* Testing Phase: Compilation of Code (error detection)

```
#read CSV file  
import pandas as pd  
df = pd.read_csv('Abhi.csv')  
print('df')
```

Implementation Phase: Final Output (no error)

SNo	Name	Regno
1)	Rahul	241801390016
2)	Nikhil	241814100007
3)	Ganesh	241814100004
4)	Umesh	241801410008
5)	Harsha	241801380024

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Design and Execution/	10		
Final Simulation/ Programming	10		
Analysis and Interpretation	10		
Overall Application of Applied and Action Learning	10		
	10		
	50		

Signature of the Student: Abhi

Name: K. Abhi, mca

Regn. No.: 241814100006
*As applicable according to the experiment.

Two sheets per experiment. Please use both sides.

241814100006

School: SATI Campus: Vizianagaram
Academic Year: 2014-15 Subject Name: DAU Subject Code: 1003
Semester: 1st Program: BTECH Branch: BCA Specialization: _____
Date: _____

Applied and Action Learning (Learning by Doing and Discovery)

Experiment:

Phase: Pseudo Code / Flow Chart / Algorithm

o code for to read csv file:

import Pandas as pd

read the csv file into a data frame

create the data frame.

o code to visualize cus file in scatter plot:

import Seaborn matplot lib pyplot numpy

variables

create a data frame with the "data" columns

name "marks" & "sin"

create a scatter plot with "name" on x-axis

marks on y-axis

display the scatter plot.

o code to visualize csv file in barplot:

import Seaborn, matplot lib pyplot and numpy

variables.

create a box plot with "name" on x-axis and

"mark" on y-axis

set the title of the plot

display the plot

* Testing Phase: Compilation of Code (error detection)

Applied and Adv.

Read csv file

```
import pandas as pd  
data = pd.read_csv('Abhinav BC(A)')  
print(data)
```

To visualize data in scatter plot

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
sns.scatterplot(x="Name", y="marks", hue="Category",  
                 data=data, alpha=1)
```

```
plt.show()
```

To visualize data plot

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

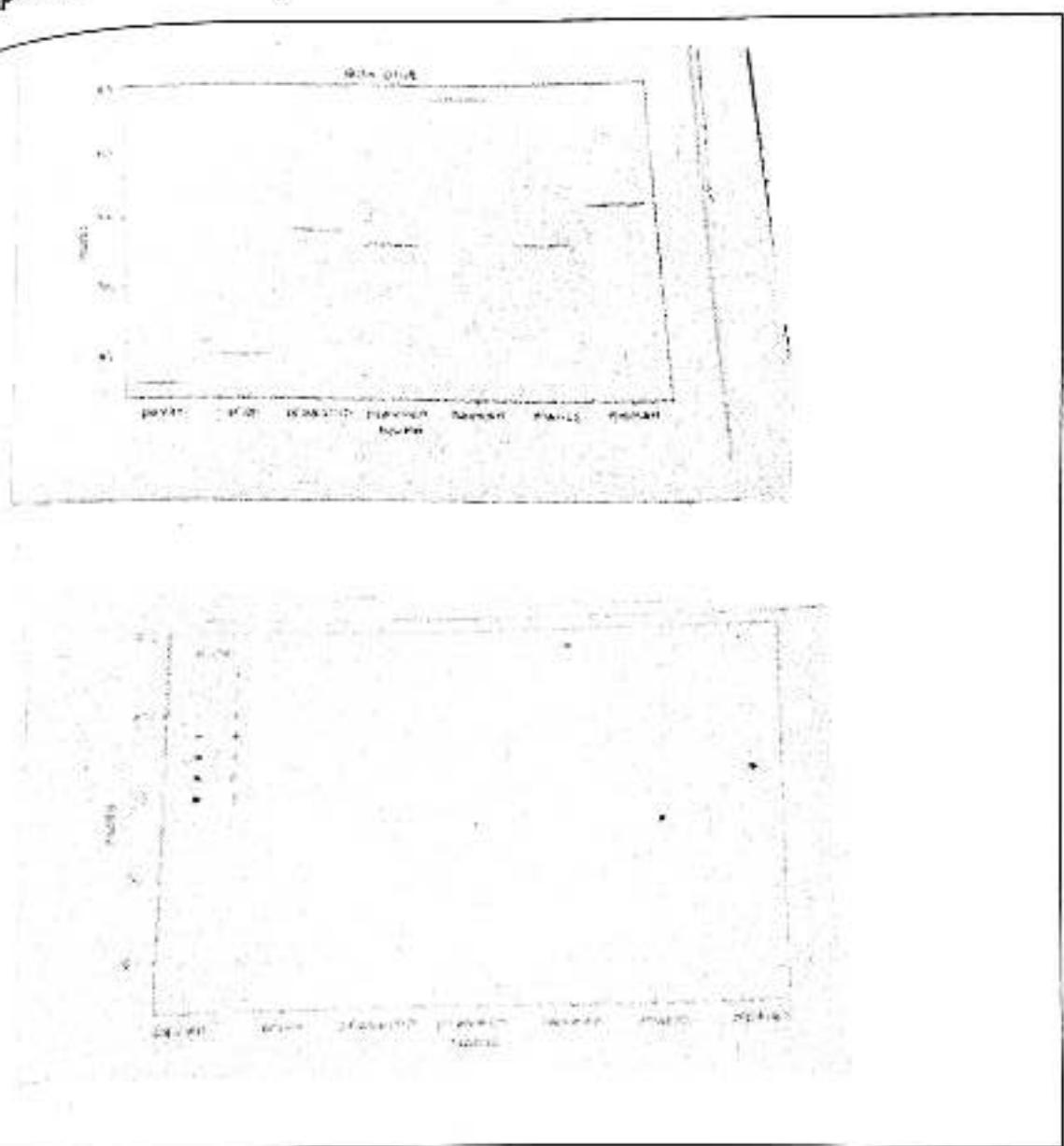
```
import numpy as np
```

```
sns.boxplot(x="name", y="marks", data=)
```

```
plt.title("Box Plot")
```

```
plt.show()
```

Implementation Phase: Final Output (no error)



ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Conceptualization	10		
Planning and Execution/ Technical Simulation/ Programming	10		
Skill and Interpretation	10		
Total of Applied and Action Learning	10		
Total	10		
Total	50		

Signature of the Student: Abhi

Name: K. Abhi

*As applicable according to the experiment.
Two sheets per experiment. Page 01 of 02

Regn. No.: 24181410006

ASSESSMENT

Applied and Action Learning

Experiment	Full Mark	Marks Obtained
Experiment - 1	50	
Experiment - 2	50	
Experiment - 3	50	
Experiment - 4	50	
Experiment - 5	50	
Experiment - 6	50	
Experiment - 7	50	
Experiment - 8	50	
Experiment - 9	50	
Experiment - 10	50	
Experiment - 11	50	

Experiment	Full Mark	Marks Obtained
Experiment - 12	50	
Experiment - 13	50	
Experiment - 14	50	
Experiment - 15	50	
Experiment - 16	50	
Experiment - 17	50	
Experiment - 18	50	
Experiment - 19	50	
Experiment - 20	50	
Average Total	50	

Signature of the Faculty

LEARNING OUTCOMES:

The Applied and Action Learning encourages Critical Thinking, Problem Solving, Idea Generation and Skill Development etc.?

able to gain knowledge with visualization good story line
ability to design plots

The Applied and Action Learning encourages Leadership, Team Work, Selection and Decision Making Capability etc.?

collaborative problem solving
building relationships working closely together on projects
decision making.

K. Abhinav
Signature of the Student

Page No. 10

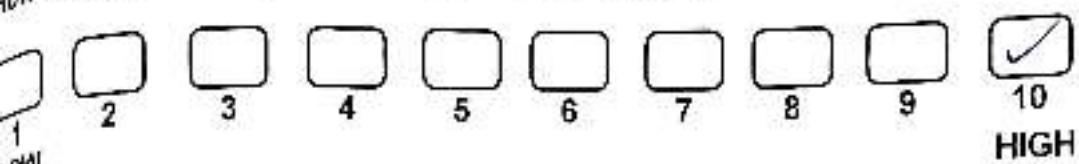
* One sheet per Learning record to be used

COURSE OUTCOMES (COs) ATTAINMENT

Expected Course Outcomes (COs):
 Refer to COs Statement in the Syllabus)

- (CO1: able to gain knowledge
 visualization with good
 story line and perform job of data
 analysis.
- (CO2: able to analyse & visualize dataset
- (CO3: able to design dashboard

Course Outcomes (COs) Attained:
 How would you rate your learning of the subject based on the specified COs?



Learning Gap (If any):

No learning gap.

Books/Manuals Referred:

E: _____

K. Abhinav
 Signature of the Student

Suggestions / Recommendations:

by the Course Faculty)

Signature of the Faculty

Page No.

*One sheet per learning record to be used



Centurion
UNIVERSITY

No. 99/2, 2nd Main
Road, Vizianagaram,
Andhra Pradesh - 535 001

**CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT
ANDHRA PRADESH**