MCA I Semester 2

Subject Name: Python Programming and Data Science-Practical **Subject Code:** MTCA23206

Type of course: Professional Core Course

Prerequisite:

- Basic understanding of programming concepts, including variables, expressions, data types, and control flow.

List of Courses where this course will be prerequisite:

- Data Analytics
- Machine learning
- Data Visualization
- Internet of Thing

Rationale: The Python lab is designed to give students hands-on coding experience, reinforcing classroom concepts through real-world problem-solving. By practicing core Python skills like control flow, data structures, and file handling, students build confidence in writing, debugging, and applying code. This practical approach prepares them for advanced areas like data science, machine learning, and software development, making them ready for future tech challenges.

Teaching and Examination Scheme:

TEACHING SCHEME				Theory Marks		Practical Marks		Total
L	T	P	C	TEE	CAT	TEP	CAP	
0	0	4	2	-	-	30	20	50

CAT: Continuous Assessment Theory comprised of CA1 and CA2 **CA1:** Continuous Assessment (assignments/projects/open book tests/closed book tests **CA2:** Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems **TEE:** Term End Examination **TEP:** Term End Practical Exam (Performance and viva on practical skills learned in course) **CAP:** Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course.

List of Practical:

Sr. No.	Problem Statements
1	Write a Python Program to Convert Celsius to Fahrenheit and vice –a-versa
2	Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal
3	Write a menu driven program to perform the following operations on strings using string built in functions. • Find the frequency of a character in a string. • Replace a character by another character in a string. • Remove the first occurrence of a character from a string. • Remove all occurrences of a character from a string.
4	Write a program in python to find out the maximum and minimum number out of three user entered numbers.
5	Write a Python program to check if the number provided by the user is an Armstrong number
6	Write a function that takes a list of strings and returns a new list that contains capitalized strings.
7	Write a function called middle that takes a list and returns a new list that contains all but the first and last elements that should be removed. So middle([1,2,3,4]) should return [2,3].
8	Write a function which breaks a string into individual letters.
9	Program to print the multiples of 3 and their sum(in the range 10 to 30).
10	Write a function which takes a list of strings and concatenates the elements
11	Consider a tuple t1= (1,2,5,7,9,2,4,6,8,10). Write a program to perform following operations: a. Print contents of t1 in 2 separate lines such that half values come on one line and other half in the next line. b. Print all even values of t1 as another tuple t2. c. Concatenate a tuple t2= (11,13,15) witht1. d. Return maximum and minimum value from t1..

12	Write a function that takes a list and returns a new list with distinct elements from the first list.
13	Write a python Program to Find the Minimum Difference Between the Index of Two Given Elements Present in List
14	Python Program to Find the Minimum Index of a Repeating Element in List
15	Python Program to Left Rotate a List by R Time.
16	Python Program to Put Even and Odd elements in a List into Two Different Lists
17	Write Function to find the largest item from a given list.

18	**List Comprehension**
	• Write a list comprehension to create a list where an element is a tuple of 3 elements x, y and z.
	  ○ The value of x is between 1 to 4.
	  ○ The value of y is between 2 to 5
	  ○ The value of z is between 5 to 8.
	Also, list should contain only those tuples as element where x + y > z
	• Write a list comprehension to make a list from the following list names with all the strings starting with 'c'.
	  ○ names = ['Ch','Dh','Eh','cb','Tb','Td']
	• create a new list using list comprehension from the range 1 to 100 such that the new list contains the square of the number and the square should be even numbers.
	• convert kilometers into feet using list comprehension.
	  ○ kilometer = [37.2, 39.5, 27.3, 17.8]
	• create a transpose of the matrix using list comprehension.
	  ○ matrix = [[10,11,12],[13,14,15],[16,17,18]]
	• Create a new list from cars where the new list will have all the cars except "Maruti". Replace "Maruti" with Mahindra.
	  ○ cars = ["Toyota", "Hyundai", "Maruti", "Tesla", "Tata", "Honda"]

19	**Dictionary** 1. Create a dictionary with the employee details and retrieve the value upon giving the keys. 2. Create a dictionary and find the sum of values.(Use eval & sinput method). 3. Create a dictionary from the keyboard and display the elements. 4. Create a dictionary with cricket player's names and scores in a match. Also retrieve runs by entering the player's name. 5. Show the usage of for loop to retrieve elements of dictionaries. 6. Find the number of occurrences of each letter in a string using dictionary 7. Sort the elements of a dictionary based on a key or values.(use lambda function) 8. Convert the elements of two lists into a key-value pair of dictionaries. 9. convert string into key value pairs and store them into a dictionary.
20	**lambda function:** 1. Write a lambda function to find the modulo of a given number. 2. Write a lambda function to calculate the square of a given number. 3. Write a lambda function to concatenate two strings. 4. Write a lambda function to find the square root of a number.
21	**Regular Expressions** 1. Create a regular expression to retrieve all words starting with 'a' in a given string. 2. Create a regular expression to retrieve all words starting with a numeric digit. 3. Create a regular expression to retrieve all words having 5 characters length. 4. Create a regular expression to retrieve all words having length of at least 4 characters. 5. Create a regular expression to retrieve all words having length of 3 or 4 or 5 characters. 6. Create a regular expression to retrieve only single digits from a string. 7. Create a regular expression to retrieve the last word of a string, if it starts with t. 8. Create a regular expression to retrieve the phone number of a person from a string. 9. Create a regular expression to retrieve birth date from a string. 10. Create a regular expression to search whether a given string is starting with 'He' or not. 11. Create a regular expression to search for a word at the ending of a string by ignoring the case. 12. Create a regular expression to retrieve the timing s either 'am' or 'pm'.
	File and Exception Handling:
22	Write a program to open a file content.txt in read mode and read the content of the file.

23	Write a function that reads a file file1 and copies only alternative lines to another file file2. Alternative lines copied should be the odd numbered lines.
24	Write a program to open a file content.txt in write mode and write data, the file should take input until it finds @ character.
25	Consider a student object with id, name, and percentage and create a class. Import this class and use pickle to dump several objects in a file namedStudent.dat. No. of objects should be taken as input. Read content from Student.dat using pickle.
26	Write a program to search for city names in the binary file named cities.bin and display the record number that contains the city name. Write a program to update or modify a record in a binary file (cities.bin). Write a program to delete a specific record from a binary file. (cities.bin). Write a program to randomly access a record from a binary file (cities.bin)
27	Write a program to create a phone book with names and phone numbers. Store data in a binary file named Phonebook.dat. Write a menu driven program to access data from Phonebook.dat. Menu Should be a. To display all the entries b. To display Phone numbers c. Modify an entry d. Exit
28	Write a Python program to handle a Zero Division Error exception when dividing a number by zero.
29	Write a program that reads a list of integers from the user and throws an exception if any numbers are duplicates.
30	Write a program to enter a specific number say n, the program should not terminate until the user enters specific number n. It should generate exception TooSmallValue if the value < n and generate exception TooLargeValue if value > n.
31	Write a program to accept positive digits. Define a user-defined exception to check whether the value given is a valid positive digit or not. If the value is not numeric the exception Not_Suitable_value_exceptions should be triggered and handled.
32	Write a user-defined exception to generate a message if the balance of the given user is less than 2000 INR in his bank account. Use a dictionary to maintain account data with username and balance.

<table>
<tr><td>33</td><td>Write a program, which accepts marks of a student (between 0 to 100) and checks whether it is within the range or not. If it is within the range then it displays "marks entered successfully", if not then it throws the exception of user defined class "MarksOutOfRangeException"</td></tr>
<tr><td>34</td><td>Write a program to generate TypeError, NameError, and ZeroDivision error and handle.</td></tr>
<tr><td>35</td><td>Using the Numpy module, write a menu driven program to do the following.
a. Create an array filled with 1's.
b. Find maximum and minimum values from an array
c. Dot product of 2 arrays.
d. Reshape a 1-D array to 2-D array.</td></tr>
<tr><td>36</td><td>Write a program that makes use of a function to display sine, cosine, polynomial and exponential curves.</td></tr>
<tr><td>37</td><td>Student Data (Student.csv)
<table>
<tr><th>Id</th><th>Name</th><th>Age</th><th>Marks</th><th>Branch</th><th>Stipend</th></tr>
<tr><td>21</td><td>Tarang</td><td>22</td><td>90</td><td>MCA</td><td>15000</td></tr>
<tr><td>22</td><td>Sanjay</td><td>30</td><td>80</td><td>MCA</td><td>20000</td></tr>
<tr><td>23</td><td>Ram</td><td>21</td><td>87</td><td>IT</td><td>6000</td></tr>
<tr><td>24</td><td>Shyam</td><td>23</td><td>76</td><td>MCA</td><td>8000</td></tr>
</table>
Create a DataFrame by reading records from the Student.csv file ad solve following queries:
1. Display the name of the student with the maximum stipend.
2. Display the names of students whose age is 25.
3. Display the total number of records in the DataFrame.
4. Display the total number of columns in the DataFrame.
5. Display the average marks of all students.
6. Display the names of students who belong to the "MCA" branch.
7. Sort the DataFrame by Marks in descending order and display the result.
8. Display the names of students who receive a stipend greater than 10,000.
9. Find and display the total stipend given to MCA students.
Display the names of students with marks greater than 85 and branch "IT".</td></tr>
</table>

38	1. Find and display the sum of salaries of employees with less than 10 years of experience. 2. Display the name and department of employees whose experience is between 5 and 15 years. 3. Display the highest-paid employee in each department. 4. Filter and display employees whose names start with 'R'. 5. Calculate and display the average age of employees in each department. 6. Display the employee(s) with the longest tenure.
39	Build a model to classify whether a person earns more or less than 50K per year based on features like age, education, and hours worked per week. Dataset : https://archive.ics.uci.edu/dataset/2/adult
40	Create a model to classify if an email is spam or not based on features like the presence of certain keywords and email length. Dataset : https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv
41	Build a regression model to predict house prices based on features like the number of bedrooms, square footage, location, and age of the house. Evaluate the model using RMSE. Dataset: https://www.kaggle.com/datasets/camnugent/california-housing-prices.
42	Create a regression model to predict car prices based on features like engine size, horsepower, mileage, and brand. Evaluate the model's performance. Dataset : https://www.kaggle.com/datasets/toramky/automobile-dataset

Suggested Specification Table with Marks (Practical):

%Distribution of Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	20	30	30	0	0

Legends: R: Remembrance, U: Understanding; A: Application, N: Analyze, E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

Sr. No.	Title of book /article	Author(s)	Publisher and details like ISBN	Year of publication	Publication Edition
1	Think Python	Allen Downey	O'Reilly	2015	2nd edition
2	An introduction to Python for absolute beginners	Bob Dowling, Cambridge Univ.	Khanna Publishing House	-	-
3	Introduction to Computation and Programming using Python	John Guttag	PHI India.	2016	-

Course Outcomes:

Sr. No.	CO Statement After learning this subject, students will be able to	Marks % Weightage
CO-1	Write programs using basics of Python, including data types, variables, control flow, and working with mutable and immutable objects	20
CO-2	Write Python functions to facilitate code reuse and handle exceptions	25
CO-3	Implement file handling operations in Python, including reading/writing text and binary files, and utilizing modules like pickle for data serialization	10
CO-4	Leverage regular expressions for effective pattern matching and string manipulation in Python to solve practical problems	15
CO-5	Employ Python libraries (Pandas, Matplotlib) for data analysis, visualization, and the development of classification and regression models	30

Mapping with POs:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PO 13
CO-1	3	2	3	3	1	2	2	1	2	2	1	1	2
CO-2	3	2	3	3	1	3	2	2	2	3	2	2	1
CO-3	3	2	3	3	1	2	1	2	3	1	1	1	1
CO-4	2	3	3	3	1	3	2	2	3	2	2	2	1
CO-5	3	3	3	3	1	3	3	2	3	2	2	2	1
Rationale*													

Rationale*: Explaining why it is matching this particular program outcome