

Programming with Python

Generic Elective (All Streams) except Computer Science

Semester- I

S. No.	Unit Name	Chapters	References	Weeks
1.	Unit 1 Introduction to Programming	2 1 (except 1.5)	[2] [1]	1-2
2.	Unit 2 Creating Python Programs	2 (2.1 to 2.3), 3	[1]	3-7
3.	Unit 3 Built-in data structures	6 (6.1, 6.2), 7 (except 7.1.11)		8-12
4.	Unit 4 File and exception handling	9 (except 9.2 and 9.5)		13-15

References

1. Taneja, S., Kumar, N., *Python Programming- A modular Approach*, Pearson Education India, 2018.
2. Balagurusamy E., *Introduction to Computing and Problem Solving using Python*, 2nd edition, McGraw Hill Education, 2018.

Additional References

1. Brown, Martin C., *Python: The Complete Reference*, 2nd edition, McGraw Hill Education, 2018.
2. Guttag, J.V. *Introduction to computation and programming using Python*, 2nd edition, MIT Press, 2016.

Practical List

1. WAP to find the roots of a quadratic equation
2. WAP to accept a number 'n' and
 - a. Check if 'n' is prime
 - b. Generate all prime numbers till 'n'

c. Generate first 'n' prime numbers

This program may be done using functions

3. WAP to create a pyramid of the character '*' and a reverse pyramid

```
*
***
*****
*****
*****

*****
*****
*****
***
*
```

4. WAP that accepts a character and performs the following:

- print whether the character is a letter or numeric digit or a special character
- if the character is a letter, print whether the letter is uppercase or lowercase
- if the character is a numeric digit, prints its name in text (e.g., if input is 9, output is NINE)

5. WAP to perform the following operations on a string

- 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.

6. WAP to swap the first n characters of two strings.

7. Write a function that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string then it should return -1.

8. WAP to create a list of the cubes of only the even integers appearing in the input list (may have elements of other types also) using the following:

- 'for' loop
- list comprehension

9. WAP to read a file and
- Print the total number of characters, words and lines in the file.
 - Calculate the frequency of each character in the file. Use a variable of dictionary type to maintain the count.
 - Print the words in reverse order.
 - Copy even lines of the file to a file named 'File1' and odd lines to another file named 'File2'.
10. Write a function that prints a dictionary where the keys are numbers between 1 and 5 and the values are cubes of the keys.
11. Consider a tuple $t1=(1, 2, 5, 7, 9, 2, 4, 6, 8, 10)$. WAP to perform following operations:
- Print half the values of the tuple in one line and the other half in the next line.
 - Print another tuple whose values are even numbers in the given tuple.
 - Concatenate a tuple $t2=(11,13,15)$ with $t1$.
 - Return maximum and minimum value from this tuple
12. WAP to accept a name from a user. Raise and handle appropriate exception(s) if the text entered by the user contains digits and/or special characters.

Course - B.Sc. (Physical Science) Computer Science

Paper : Python Programming for Data Handling

Semester- III

	TOPICS/UNITS	Chapter	Ref
Week 1 to 5	Unit 1 (15 Hours) Introduction to Python Programming, Basic Constructs, and Python Built-in Data Structures: Introduction to Python programming language, Basic syntax, variables, and data types in Python, Functions and modular programming; Conditional statements (if, elif, else); Looping structures (for and while loops); Mutable and Immutable Data Structures, Strings- Indexing, slicing, traversal, operations; Lists-indexing, slicing, traversal, operations; tuples, dictionaries, and sets and their operations in Python.	Ch. 1, Ch. 2, Ch. 3, Ch. 6, Ch. 7	[1]
Week 6 to 7	Unit 2 (5 Hours) File Handling: Opening, reading, writing, and closing files; File modes and file object methods; Reading and writing text and binary files; Working with CSV files	Ch. 9:9.1,9.2 Online reference 1 [csv.reader() and csv.writer(), dictReader() and dictWriter()] Online reference 2 [7.2. Reading and Writing Files, 7.2.1]	[1]
Week 8 to 12	Unit 3 (15 Hours) Designing GUI Applications with Tkinter (15): What is Tkinter? Creating a Tkinter window, Layout managers, Tkinter widgets - Entry, Spinbox, Combobox, Checkbutton, Text, Button, LabelFrame; Implementing the application - LabelInput class, building of form, adding LabelFrame and other widgets, retrieving data from form, resetting form, building our application class.	Ch. 1, Ch. 2, Ch. 3 (Till Reset Function)	[2]
Week 13 to 15	Unit 4 (10 Hours) Combining Python file handling and Tkinter: Creating a simple Tkinter application, Reading and Writing to csv files in a Tkinter application.	Ch. 3 (The Save callback)	[2]

References

1. Taneja S., Kumar, N. Python Programming- A modular approach, 1st Edition, Pearson Education India, 2018,
2. Moore, Alan D. Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter. Packt Publishing Ltd, 2021.

Additional References:

1. Guttag, J.V. Introduction to computation and programming using Python, 2nd edition, MIT

Online references/material:

1. <https://docs.python.org/3/library/csv.html>
2. <https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files>

Suggestive Practice Questions:

Installing and setting up Python and relevant libraries; Python development environments (e.g., Anaconda, Jupyter Notebook)

1. Write a Python program to calculate the factorial of a number.
2. Write a Python program to generate prime numbers between 1 to n, where n is provided as input by the user.
3. Write a Python program to find the sum and average of numbers in a given list.
4. Given two sets, set1 and set2, write a Python program to find their union, intersection and difference.
5. Given a list of numbers, write a Python program to count the number of times an element occurs in a list and create a dictionary with *element:count* as *key:value* pairs.
6. Write a Python program to swap the first two and last two characters in a given string.
7. Write a Python program to create a text file having names of ten Indian cities.
8. Write a Python program to create a text file having atleast five lines about your college using `writelines()` function.
9. Write a Python program which reads the data from two input files having Employee Names and merges them into one output file.
10. Write a Python program to count the number of vowels in a file and write the *vowel : count* in a dictionary.
11. Write a Python program to create a CSV file having student data: RollNo, Enrollment_No, Name, Course, Semester.
12. Write a Python program library to read the CSV file created in the above program and filter out records of II semester students.
13. Write a Python program using tkinter library to create a GUI to enter registration details for an event.
14. Write a Python program using tkinter library to create a calculator to perform addition, subtraction, multiplication and division of two numbers entered by the user.
15. Write a Python program using tkinter library to create an age calculator to calculate age when DOB is entered.
16. Write a Python program using tkinter library to read student details, namely, RollNo, Enrollment_No, Name, Course, Semester, through a form and write the entered student details to a CSV file.