Faculty of Science

Bachelor of Computer Application (B.C.A.)

Semester: V

Paper: Core

Subject: Programming with Python

Course Outcomes

CO. No.	Course Outcomes	Cognitive Level
CO 1	Understand Python's fundamentals and development environments, including installation, data types, variables, operators, and input/output operations.	U, R
CO 2	Master Python's control structures, data collections, and functions, including conditional and loop statements, strings, lists, tuples, sets, dictionaries, and higher-order functions.	U, R, Ap
CO 3	Understand the importance of modular programming, creating and using predefined and user-defined modules and packages, and file and directory handling in Python.	U, Ap, C
CO 4	Comprehend procedural vs. object-oriented programming, OOP principles (encapsulation, abstraction, polymorphism, inheritance), inner classes, and exception handling.	An, Ap, C
CO 5	Learn multithreading and multiprocessing in Python, thread lifecycle methods, synchronization, and Numpy, Pandas, and Matplotlib for data handling and plotting.	U, Ap, C

Credit and Marking Scheme

	Credits	Marks		Total Marks
		Internal	External	
Theory	4	40	60	**100**
Practical	2	40	60	**100**
Total	**6**	**200**		

Evaluation Scheme

	Marks	
	Internal	**External**
Theory	3 Internal Exams of 20 Marks (During the Semester) (Best 2 will be taken)	1 External Exams (At the End of the Semester)
Practical	3 Internal Exams (During the Semester) (Best 2 will be taken)	1 External Exams (At the End of the Semester)

Bachelor of Computer Application (BCA)
V Semester

Subject: Programming with Python
Paper: Core

Content of the Course

Theory

No. of Lectures (in hours per week): 2 Hrs. per week

Total No. of Lectures: 60 Hrs.

Maximum Marks: 60

Units	Topics	No. of Lectures
I	What is Python? WHY PYTHON? History, Features - Dynamic, Interpreted, Object-oriented, Embeddable, Extensible, Large standard libraries, Free and Open source. Download &Python Installation Process in Windows, Unix, Linux, and Mac, Online Python IDLE, Python Realtime IDEs like Spyder, Jupyter Notebook, PyCharm. Rodeo, Visual Studio Code, ATOM, PyDev, etc., Data Types and Variables, Numbers, Operators Comments in Python. Input-output operation in Python, str.format().	10
II	Control Statements: Conditional control statements - if, If-else, If-elseif-else, Loop control statements- for, while, Data Structure & Collection: - String, List, Tuple, Set, Dictionary, Comparison of List, Tuple, and Set, Function in Python, types of function in Python, map, reduce, filter function. Lamda Function.	10
III	Importance of modular programming. What is a module? Types of Modules: Pre-defined, User-defined. A user defines module creation, OS, Date-time, math modules, organizing Python projects into packages, Types of packages – predefined, user-defined. Package v/s Folder, File, and Directory handling in Python.	10
IV	Procedural v/s Object-oriented programming, Principles of OOP - Encapsulation, Abstraction (Data Hiding), Polymorphism, Inheritance. Inner Classes. Exception handling and types of errors, try, except, finally, raise, and Need to Custom exceptions, Case studies, regular expression.	15
V	Multithreading and multiprocessing in Python, the Life cycle of a thread. Need to start() method , Sleep() & Join(), Synchronization -Lock class - acquire(), release() functions. Python Data Base Communications (PDBC), Introduction of Numpy, Numpy Array, Pandas data frame& MatPlotLib, Drawing plots.	15

References

Text Books:

- Mark Lutz, Learning Python
- Tony Gaddis, Starting Out with Python
- Kenneth A. Lambert, Fundamentals of Python
- James Payne, Beginning Python using Python 2.6 and Python

Reference Books:

- Python Crash Course: A Hands-On, Project-Based Introduction to Programming Edition Eric Matthes.
- The Python Language Reference Manual (version 3.2), Guido van Rossum, Drake, Jr. (Editor), ISBN: 1906966141, Network Theory Ltd, 120 pages

Suggestive digital platforms/ web links:

- www.javatpoint.com
- www.w3school.com
- www.python.org
- https://www.tutorialspoint.com/Python/index.htm

List of Practical

1. Write a program to demonstrate different number data types in Python.
2. Write a program to perform different arithmetic Operations on numbers in Python.
3. Write a program to create, concatenate print a string and access a sub-string from a given string.
4. Write a Python script to print the current date in the following format a. "Fri Oct 11
5. Write a program to create, append, and remove lists in Python.
6. Write a program demonstrating working with tuples in Python.
7. Write a program demonstrating working with dictionaries in Python.
8. Write a Python program to find the largest of three numbers.
9. Write a Python program to construct the following pattern, using a nested for loop

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

10. Write a Python script that prints primenumberslessthan20.
11. Write a Python program to define a module to find Fibonacci Numbers and import the module to another program.
12. Write a Python program to define a module and import a specific function in that module to another program.
13. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
14. Write a Python class to convert an integer to a Roman numeral.
15. Write a Python class to reverse a string word by word.