Programme	B. Sc. Computer Science				
Course Title	Data Visualisation using Python				
Type of Course	**Minor**				
Semester	III				
Academic Level	200-299				
Course Details	Credit	Lecture per week	Tutorial per week	Practical per week	Total Hours
	4	3	-	2	75
Pre-requisites	Have an understanding about algorithms and flowchart				
Course Summary	This course explores the versatility of Python language in programming and teaches the application of various data structures using Python.				

Course Outcomes (CO):

CO	CO Statement	Cognitie Level*	Knowledge	Evaluation Tools used
CO1	Discuss the basic concepts of Python programming including datatypes and operators	U	C	Instructor-created exams / Quiz
CO2	Apply problem-solving skills using different control structures and loops	Ap	P	Coding Assignments/ Code reading and review
CO3	Discuss various data structures and operations using Python	U	C	Instructor-created exams / Quiz / Short coding-based questions
CO4	Apply modular programming using functions	Ap	P	Coding Assignments / Lab exercises
CO5	Design Python programs and implement data	C	P	Mini Project / Coding Assignment /

	visualization techniques			Practical Demonstration
* - Remember (R), Understand (U), Apply (Ap), Analyse (An), Evaluate (E), Create (C) # - Factual Knowledge(F) Conceptual Knowledge (C) Procedural Knowledge (P) Metacognitive Knowledge (M)				

Detailed Syllabus:

Module	Unit	Content	Hrs	Marks
I	**Introduction to Python**		**12**	17
	1	Features of Python, Different methods to run Python, Python IDE	2	
	2	Comments, Indentation, Identifiers, Keywords, Variables	2	
	3	Standard Data Types	2	
	4	Input Output Functions, Import Functions, range function	1	
	5	Operators and Operands, Precedence of Operators, Associativity	2	
	6	Type Conversion, Multiple Assignment	1	
	7	Expressions and Statements, Evaluation of Expressions	1	
	8	Boolean Expressions	1	
	Control Structures		**12**	19
	9	Decision Making- if statement, if…else statement, if…elif…else statement, Nested if statement	5	
II	10	Loops - for loop, for loop with else, while loop, while loop with else, Nested Loops	5	
	11	Using indentation in Python to define code blocks	1	
	12	Control Statements- break, continue, pass	1	
	Data Structures in Python		**12**	19
	13	Working with strings and string manipulation	3	

III	14	List - creating list, accessing, updating and deleting elements from a list	2	
	15	Basic list operations	1	
	16	Tuple- creating and accessing tuples in python	2	
	17	Basic tuple operations	1	
	18	Dictionary, built in methods to create, access, and modify key-value pairs	2	
	19	Set and basic operations on a set	1	
	Functions		**9**	15
IV	20	Built-in functions - mathematical functions, date time functions, random numbers	1	
	21	Writing user defined functions - function definition, function call, flow of execution, parameters and arguments, return statement	6	
	22	Recursion. Introduction to basic Python libraries (e.g., math, random)	2	
	Hands-on Data Structures: Practical Applications, Case Study and Course Project		**30**	
Design programs from the concepts listed below. Select the topics and programs suited for your domain				
V	1	• Read input, include casting that input to the appropriate type • Select from one of several alternatives by using an if-elif or if-elif-else statement • Use the range() function in a form loop • Call and use functions residing in the math module		

		Case study: ● Design a basic calculator application in Python that can perform addition, subtraction, multiplication, and division. ● Create a Python program that retrieves weather data from an API (e.g., OpenWeatherMap) and displays it.		
	4	**Data Structures in Python** • String - Create a string , Indexing / Looping / Slicing • Lists - Create a list , Indexing /Looping / Slicing , Adding items / Modifying items / Removing items • Tuples - Create a tuple , Indexing / Looping / Slicing / Adding items to a tuple • Dictionary - Create a dictionary and access values with key / Adding a key- value pair / Adding to an empty dictionary /Modifying values in a dictionary / Removing key-value pair		
	5	**Function** • Call functions residing in the math module • Define a function for later use • Pass one or more values into a function • Return one or more results from a function • Call a function that you have defined previously		

Mapping of COs with PSOs and POs :

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PO 1	PO2	PO3	PO4	PO5	PO6	PO7
CO 1	-	-	-	-	-	-	3	1	3	3	3	1	3
CO 2	-	-	-	-	-	-	3	1	3	3	3	1	3
CO 3	-	-	-	-	-	-	3	1	3	3	3	1	3
CO 4	-	-	-	-	-	-	3	1	3	3	3	1	3
CO 5	-	-	-	-	-	-	3	1	3	3	3	1	3

Correlation Levels:

Level	Correlation
-	Nil
1	Slightly / Low
2	Moderate / Medium
3	Substantial / High

Assessment Rubrics:

- Quiz / Assignment/ Quiz/ Discussion / Seminar
- Midterm Exam
- Programming Assignments (20%)
- Final Exam (70%)

Mapping of COs to Assessment Rubrics :

	Internal Exam	Assignment	Project Evaluation	End Semester Examinations
CO 1	✓			✓
CO 2		✓		✓
CO 3	✓			✓
CO 4	✓	✓		✓
CO 5		✓	✓	

Reference Books:

1. Jose, Jeeva. Taming Python By Programming. Khanna Book Publishing, 2017. Print.
2. Downey, Allen. Think Python. Green Tea Press, 2nd ed. 2009