

Programme	B. Sc. Computer Science				
Course Code	CSC1MN102				
Course Title	Python Programming				
Type of Course	Minor				
Semester	I				
Academic Level	100-199				
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	Cognitive Level*	Knowledge	Evaluation Tools used
CO1	Understand the basic concepts of Python programming	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	Design simple Python programs to solve basic computational problems and acquire knowledge of Python's error handling mechanisms to effectively debug	Ap	P	Coding Assignments/ Exams

	programs			
CO4	Analyse the various data structures and operations on it using Python	An	P	Instructor-created exams / Case studies
CO5	Apply modular programming using functions	U	C	Instructor- created exams / Quiz
CO6	Identify the necessary Python packages in the domain and create simple programs with it	U, Ap	C, P	Coding
<p>* - Remember (R), Understand (U), Apply (Ap), Analyse (An), Evaluate (E), Create (C)</p> <p># - Factual Knowledge(F) Conceptual Knowledge (C) Procedural Knowledge (P) Metacognitive Knowledge (M)</p>				

Detailed Syllabus:

Module	Unit	Content	Hrs	Marks
I	Introduction to Python		12	20
	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	
II	Control Structures		12	20
	9	Decision Making- if statement, if...else statement, if...elif...else statement, Nested if statement	5	
	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	
III	Data Structures in Python		12	20
	13	Working with strings and string manipulation	3	
	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	<p>Programs to:</p> <ul style="list-style-type: none"> • Run instructions in Interactive interpreter and as Python Script • Perform calculations involving integers and floating point numbers using Python arithmetic operators <p>Data Structures in Python</p> <ul style="list-style-type: none"> • 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 <p>Function</p> <ul style="list-style-type: none"> • 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 		
		<p>Case study:</p> <ul style="list-style-type: none"> • Create a Todo List Manager where Users should be able to add, remove, and view tasks • Create Student Grade Tracker: Allow users to add students, add grades for subjects, and calculate average grades. 		

Mapping of COs with PSOs and POs :

	PSO 1	PSO 2	PSO 3	PSO4	PSO5	PSO6	PO 1	PO2	PO3	PO4	PO5	PO6
CO 1	-	1	2	3	1	1						
CO 2	-	1	2	3	1	1						
CO 3	-	2	2	3	1	1						
CO 4	1	1	-	-	1	-						
CO 5	1	1	2	2	1	-						
CO 6	-	1	2	2	2	1						

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	✓			✓
CO 6	✓			✓

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