Name of the Programme: MSc Integrated
Course Code: IMC- 104
Title of the Course: Programming in Python
Number of Credits: 6(4L-0T-2P)
Effective from AY: 2020-21

Prerequisites for the course:	Same as programme pre-requisites	
Objectives:	The aim of the course is to provide an exposure to solve common computing problems through programming using Python language. The course is designed with a lab component to give the student hands-on experience of the basic concepts of programming.	
Content Theory:	Introduction to computer systems and data representation: Functional units of a Computer, Characteristics of a Computer, Data representation and Storage,Evolution of Programming Languages, Compilation and Interpretation, Structured and Procedural Programming languages	3 hours
	The Problem Solving Process: – Requirement Analysis, Algorithmic Construction, Identifying Test Cases, Desk Checking, Implementation, Testing and maintenance issues, Data verification and validation.	4 hours
	Python Programming Environment: Python overview, Structure of Python program, character Set, variable declarations and data types, Program Statements, Types of Instructions, Expression Evaluation rules, Type Conversions. Managing I/O operations	4 hours
	Selection and Iterative Constructs: Writing conditions, IF-ELSE constructs Conditional operators, SWITCH,WHILE and FOR loops, Use of BREAK and CONTINUE statements. Nested Loops	9 hours
	Advance Data types: Lists, Tuples, Set, Dictionaries, Strings, Unicode, formatting strings, docString. Searching and sorting algorithms without using library functions.	6 hours
	Modular Programming: Importance of User Defined Functions, Hierarchy charts, fan-in/out, cohesion and coupling and loosely coupled modules. Fan-in – Fan-out concepts.	5 hours
	User Defined Functions: Local and Global Variables, Scoping Rules, Parameters& arguments. Function with variable arguments. Modules, packages, scope. Recursion & Recursive Functions. Recursive v/s Iterative Functions.	7 hours
	Custom Data Types and File Management: Object of a Class and basic concept of classes & OOP, Files, Exceptions in file handling.	4 hours
	Introduction to Packages: Python packages for plotting, mathematical computation& linear regression.	6 hours

Content Practical:	Suggested Lab Assignments: minimum 16 assignments and duration of carrying out each assignment 3 hrs. 1. Introduction to UNIX environment- Introduction to Fedora/Ubuntu, Basic directory and file handling commands, Editor (vi editor), man pages, installation of Python and Jupyter notebook. Programs using decision control, branch and loop control structure 1. Program to find the largest of three numbers 2. Program to print the reverse of a given number. 3. Program to check whether a given number is Armstrong or not	16 * 3 = 48 hours

	4. Program to print the prime numbers from 2 to n, where n is an input given by the user. 5. Program to print the patterns. Programs using List, Set, Tuple, Dictionary & Strings 6. Program to find the largest and smallest number in a list of integers (without using library function). 7. Program to sort a given integer list in ascending order (without using library function). 8. Program to print the sum and average of the elements of the list (without using library function). 8. Program to find the duplicate elements in the list (without using library function). 9. Program to reverse a given string and check whether it is palindrome (without using library function). 10. Program to read a string and count the number of vowels in it. 11. Program to concatenate two strings without using library functions 12. Program to arrange the list of names in alphabetical order. 13. Program to find the union, interaction and difference between two sets. 14. Program to take a sentence as an input from the user and compute the frequency of each letter. Make use of dictionary type to maintain the count. 15. Programs using functions & Recursion. 16. Write functions for addition, subtraction and multiplication of two matrices. Each function has two matrices as parameters and returns the result. 17. Program to print the Fibonacci series using recursion. 18. Program to find the GCD of two numbers using recursion. 19. Program to solve Tower of Hanoi Programs user-defined data types & file handling 20. Program to store the item number, name, rate and quantity of 'n' items in a custom data type, where n is given as input by the user. Display the total value inventory items. 21. Program to store employee details in a Custom data type. The data should include employee ID, name, salary, and date of joining. The date of joining should be stored in a structure. The program should perform the following operations based on a menu selection a) Display the details of the employees who have more than 5 years of experience with the company. b) Increase the salaries according to the pay scale rules 22. Program to create a custom data type of Student with fields Roll No, Name, course, and Total Marks. Read the data from the user and store them in a file. Write a function to display the Roll No, name of the student who has secured the highest marks. 23. Program to count the number of characters in a file. 24. Program to search for a particular word in a file. 25. Program to handle various file exceptions. 26. Program to implement linear regression method. 27. Program to plot graphs.	
Pedagogy:	Lectures/Practical/ tutorials/assignments/self-study.	

References/ Readings	1. Taneja Sheetal, Kumar Naveen , —Python Programming - A modular approach, Pearson 2017 2. Guttag John V., —Introduction to Computation and Programming using Python, MIT Press, 2nd Edition 2016. 3. Maureen Sprankle, Jim Hubbard — Problem Solving and Programming Concepts, Pearson, 9th Edition 2012
Course Outcomes	1. Analyze a given problem and develop a Python program to solve it. 2. Identify test cases for a given problem. 3. Understand, test, trace programs written in Python language. 4. Working with python Standard Libraries,User Defined Functions,Custom Data Types and File Management and Packages