

OLMDS510	Python Programming		L	T	P	C
			3	1	0	4
Pre-Requisite	NIL	Syllabus Version				
		1.0				
Course Objectives:						
1. To introduce core programming basics required for data science using Python language. 2. To read and write simple Python programs. 3. To develop Python programs with conditionals and loops. 4. To use Python data structures – lists, tuples, dictionaries. 5. To introduce the important data science modules NumPy, SciPy and Matplotlib. 6. To introduce the input/output with files in Python and statistical processing of a data using Pandas.						
Course Outcome:						
At the end of the course students will be able to: 1. Write, execute simple Python programs. 2. Develop a Python program using functions. 3. Manipulate 1-d, 2-d and multidimensional data using Python. 4. Read and write data from/to files in Python. 5. Develop algorithmic solutions to data science related problems.						
Module:1	Algorithmic Problem Solving					
Algorithms, building blocks of programs (statements, state, control flow, functions); algorithmic problem solving; iteration, recursion. Illustrative problems: finding minimum in a list, guess an integer number in a range, factorial of a number.						
Module:2	Data, Expressions, Statements in Python					
Python Strengths and Weakness; Installing Python; IDLE - Spyder – Jupyter; Mutable and Immutable Data Types, Naming Conventions; String Values; String Operations; String Slices; String Operators; String functions – split, join, chr, ord; Numeric Data Types; Arithmetic Operators and Expressions; Comments in the Program; Understanding Error Messages.						
Module:3	Formatting, Operators, program control and Collections data types					
Control Flow and Syntax; Indenting; The if statement; Relational Operators; Logical Operators; Bit-wise Operators; The while Loop – break and continue statements; The for Loop; List; Tuples; Sets; Dictionaries; Sorting Dictionaries; List Comprehension, Strings.						
Module:4	Functions and Modules in Python					
Functions - Introduction; Defining your own functions; parameters; local and global scope; passing collections to a function; variable number of arguments; passing functions to a function; Lambda function; map; filter; Modules: Introduction; Standard Modules – sys, math, time.						
Module:5	Python Modules for Data Science – I					
NumPy arrays – 1-d, multidimensional arrays and matrices; Mathematical operations with arrays; Slicing and addressing arrays; Boolean masks; Difference between lists and arrays; SciPy – Scientific Computing library of Python – Introduction, Basic functions, Special functions, scipy.integrate, scipy.optimize, scipy, interpolate.						
Module:6	Python Modules for Data Science – II					
Python Plotting: PyPlot – Basic Plotting; Logarithmic Plots; Plots with multiple axes; Matplotlib – interactive functions 3d plotting; Pandas – Introduction, DataFrame, Reading and writing CSV, XLS files, Working with missing data, categorical data, data visualization with pandas.						
Module:7	Error Handling in Python					
Handling IO Exceptions, Metadata, Errors, Runtime Errors, Exception Model.						
	Total Lecture hours:					120 hours

Text Book(s)	
	- David J. Pine, Introduction to Python for Science and Engineering, CRC Press, 2019. - Jake vander Plas, Python Data Science Handbook – Essential Tools for Working with Data, O'Really Media, 2017.
Reference Book(s)	
	- Robert Johansson, Numerical Python – Scientific Computing and Data Science Applications with NumPy, SciPy And Matplotlib, Apress, 2019. - Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016. - Nelli, F., Python Data Analytics: with Pandas, NumPy and Matplotlib, Apress, 2018.
Mode of Evaluation: Quiz, Assignment, Mid-Term and FAT.	
List of Tutorials(Indicative)	
1. Python Program Environment, IDLE, Jupyter, Spyder environments <i>First Basic Experiment(s):</i> (i) "Hello World!" Program in IDLE, Jupyter, Spyder Environments. (ii) Program(s) to demonstrate the Python data types	
2. Python Operators, Expressions and Flow Controls <i>Simple Experiment(s):</i> (i) Program to demonstrate the Python operators and their order of preference. (ii) Program to add/multiply/divide two numbers (iii) Program to verify whether a given number is even or odd <i>Perfection:</i> Program to verify whether a given number is Armstrong number or not. A number is said to Armstrong number if sum of the cubes of individual digits of that number is equal to the number itself. Viz., $153 = 1^3 + 5^3 + 3^3$	
3. Python Lists, Tuples, Dictionaries & Sets <i>Simple Experiment:</i> Write a Python program which demonstrate the use of Lists, Tuples Dictionaries and Sets. This program should accepts the elements into various types and perform the other operations such as append, copy, extend, pop, remove operations.	
4. Python Functions, Modules and Packages <i>Simple Experiment(s):</i> Write a function file which accepts a set of numbers and displays the largest among them <i>Perfection:</i> Write a function which accepts a number 'n' and list the first 'n' Fibonacci numbers <i>Challenging:</i> Create a own module in Python which includes functions such as greeting() which greets a welcome message to user. This module should also contain some variables and functions which finds the maximum among the two given numbers.	
5. Array and Matrix Manipulation in Python <i>Simple Experiment:</i> Write a Python program demonstrating the NumPy matrix operations such as accepting two matrices finding the dimension, adding the two matrices <i>Perfection:</i> Write a Python program which accepts a matrix A of order m x p another matrix B of order p x n and checks whether the matrix multiplication is possible or not. If possible then finds matrix multiplication and displays it to user.	
6. Data Manipulation – SciPy Module <i>Simple Experiment:</i> Write a Python program to find the det, inv, eigenvalues and eigenvectors of a matrix using corresponding SciPy module functions <i>Challenging:</i> Create a data set consisting of time series observations of an experiment. Using the interpolation techniques of SciPy module form an	

interpolating polynomial and use it to estimate the experimental values for intermediate values.			
7. Data Visualization in Python – PyPlot Module <i>Compare:</i> Given the examination scores of students of three different classes for the same subject taught by different professors, display them visually to aid comparison of pass percentage, A grades etc.			
8. Data Manipulation using Pandas – Exploring a Dataset and Analysing a Dataset <i>Simple Experiments:</i> Create a data frame consists of five countries, their capitals, area of the country, population. The program should also print the description of the data frame and finally save this data frame to a csv file. <i>Challenging:</i> Write a Python program demonstrating the Pandas indexing capabilities, identifying the null values in the dataset and filling them with or dropping them from the dataset. Also demonstrate the merging, joining and concatenating data frames using Pandas.			
9. Descriptive Statistical Analysis – Evaluation, Plotting and Interpretation <i>Linear Regression:</i> Read a data frame in csv/xls format containing the weather data such as pressure, min temp, max temp, humidity, rainfall. Using the Pandas, Matplotlib and SciPy plot the scatter plots and develop a linear interpolation between rainfall with all other parameters and evaluate the statistical significance of the model.			
10. Evaluation of Probability using various Distributions Functions <i>Simple Experiments:</i> Write Python programs to generate a normal distribution, binomial distribution and Poisson distribution using Python and visualize them. <i>Challenging:</i> Write Python program to check the normality of a dataset, which a foremost important test, required to determine whether to apply parametric tests or nonparametric tests on the given test. These tests include Histogram, Quantile-quantile plot, Shapiro-Wilk test, D'Agotino's K-squared test, Anderson-Darling test			
11. Linear and Nonlinear Regression in Python <i>Simple Linear Regression:</i> Write a Python program to implement the Simple Linear Regression model to predict the wine quality using the physicochemical and sensory variables by using Scikit-Learn module and estimate the statistical significance of the model. <i>Nonlinear Linear Regression:</i> Write a Python program to predict the price of oil (OIL) from indicators such as the West Texas Intermediate (WTI) price, Henry Hub gas price (HH), and the Mont Belvieu (MB) propane spot price. Data is available for OIL, WTI, HH, and MB from the years 2000 to 2016 at the link https://apmonitor.com/me575/uploads/Main/oil_data.txt . The OIL is related with WTI, HH and MB nonlinearly as follows: $OIL = A (WTI^B) (HH^C) (MB^D)$			
12. Decision Trees and Time Series Analysis in Python Programs to illustrate the use of decision trees in machine learning to develop the decisions and their possible consequences. In this experiment we will use the dataset related breast cancer to predict the breast cancer spread using decision trees.			
Mode of Evaluation: Quiz and FAT			
Recommended by Board of Studies		28.10.2021	
Approved by Academic Council		No.65	Date 17-03-2022