Subject Code: 01CT1309

Subject Name: Programming with Python

B. Tech. Year – II (Semester III)

Objective: The objective of this course is to describe the core syntax and semantics of Python programming language and make student understand the need for working with the strings and functions. Further, this course will include Object-oriented Programming concepts and develop the ability to write GUI and database applications in Python using NumPy, pandas and matplotlib python libraries.

Credits Earned: 02 Credits

Course Outcomes: After completion of this course, student will be able to:

1. Understand flowcharts and algorithms that help to develop a creating a logical foundation for any real problem (Understand).
2. Understand how to use various data structures, modules, primitive datatypes and user defined functions (Understand).
3. Analyse how various control flow statements (such as iterative, transfer and conditional statements) are used in a complex problem (Analyse).
4. Apply the knowledge of decisional control statements to deal with pre-processors, functions, modules, and file handling to enhance coding skills. (Apply)
5. Create a model for analysis of data using NumPy, pandas and matplotlib (Create).

Pre-requisite of course: Basic knowledge of Programming, Object Oriented Concepts

Teaching and Examination Scheme:

Teaching Scheme (Hours)			Credits	Theory Marks			Tutorial / Practical		Total Marks
				E	I		V	T	
Theory	Tutorial	Practical		ESE	IA	CSE	Viva	Term Work	
0	0	4	2	00	00	00	50	50	100

Contents:

Unit	Topics	Contact Hours
1	**Introduction:** Introduction to Python programming language; data and expressions: literals; variables and identifiers; operators; expressions and data types; control structures, Boolean expressions	5
2	**Statements, List, Dictionaries, Sets and Functions:** Selection Control, Iterative Control, Lists: List Structures; Lists in Python; Iterating over Lists in Python, Dictionary, Set and Tuple in Python Program routes; Calling Value Returning Functions; Calling Non-value Returning Functions; Parameter Passing; Keyword and Default Arguments in Python	11
3	**Modules and OOP, GUI Programming and File Handling and Database Connectivity:** Python modules, classes and OOP, exception handling, string processing and regular expression, introduction to GUI programming, controls, event handling, file handling: opening, reading and writing text files, database connectivity with python	12
4	**Introduction to Python Libraries:** introduction to data science, exploratory data analysis and data science process. motivation for using python for data analysis, introduction of Jupyter Notebook, Overview of NumPy, pandas, matplotlib, SciPy	4
5	**Getting Started with Pandas:** Arrays and vectorized computation, introduction to Pandas data structures, essential functionality, summarizing and computing descriptive statistics, data loading, storage and file formats, reading & writing data in text format, web scraping, binary data formats, interacting with web API’s, interacting with databases, data cleaning and preparation, handling missing data, data transformation.	8
6	**Getting Started with NumPy:** Array, NumPy array, indexing and slicing, operations on arrays, concatenating arrays, reshaping arrays, splitting arrays, statistical operations on arrays, loading arrays from files, saving numpy arrays in files on disk	8
7	**Signal Processing using Python:** continuous and discrete time elementary signals, basic signal processing operations, convolution and correlation, sampling and reconstruction, LTI systems, frequency domain analysis: Z-transform and Fourier transform	8
	Total Hours	56

Suggested Text books / Reference books:

1. Mark Lutz, “Learning Python”, O Reily, 4th Edition, 2009, ISBN: 978-0-596-15806-4
2. Mark Lutz, “Programming Python”, O Reily, 4th Edition, 2010, ISBN 9780596158118
3. Tim Hall and J-P Stacey, “Python 3 for Absolute Beginners”, 2009, SBN:9781430216322
4. Magnus Lie Hetland, “Beginning Python: From Novice to Professional”, 2nd Edition, 2009, ISBN:978159059982
5. C. Dierbach, “Introduction to Computer Science Using PYTHON: A Computational Problem-Solving Focus” 1st edition, Wiley, 2015. ISBN 978-0470555156.
6. Yashavant Kanetkar, “Let Us Python” 1st edition, BPB Publishers, 2019. ISBN 978-9388511568.
7. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist” 2nd edition, O’Reilly, 2015. ISBN 978-9352134755.
8. Martin C. Brown, “Python: The Complete Reference” 1st edition, McGraw-Hill, 2001. ISBN 978-9387572942.
9. McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython. 2nd edition. O’Reilly Media.
10. O’Neil, C., & Schutt, R. (2013). Doing Data Science: Straight Talk from the Frontline O’Reilly Media
11. José Unpingco, Python for Signal Processing, Springer, ISBN 13 9783319013428,

Suggested Theory distribution:

The suggested theory distribution as per Bloom’s taxonomy is as per follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process.

Distribution of Theory for course delivery and evaluation					
Remember	Understand	Apply	Analyze	Evaluate	Create
15%	20%	25%	15%	15%	10%

Suggested List of Experiments:

Minimum 24 experiments to be performed during the semester

1. Write a program to demonstrate different number datatypes in python.
2. Write a program to perform different arithmetic operations on numbers in python.
3. Write a program to create, concatenate and print a string and accessing substring from a given string.
4. Write a python script to print the current date in following format “Sun August 01 10:10:10 IST 2021”
5. Write a python program to create, append and remove lists in python.

6. Develop programs to understand the control structures of python.
7. Write a program to demonstrate working with tuples in python.
8. Write a program to demonstrate working with dictionaries in python.
9. Write a python program to find largest of three numbers.
10. Write a python program to convert temperature to and from Celsius to Fahrenheit.
11. Write a python program to print prim numbers less than given number.
12. Write a python program to find factorial of a number using recursion
13. Write a python program to that accepts length of three sides of a triangle as
14. inputs. The program should indicate whether or not the triangle is a right-angled triangle (use Pythagorean theorem)

15. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
16. Write a python program to define a module and import a specific function in that module to another program.
17. Write a python Program to call data member and function using classes and objects.
18. Write a program to read 3 subject marks and display pass or failed using class and object.
19. Write a program to validate PAN card number and Email ID.
20. Write a GUI program to create Tic-tac-toe in python.
21. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first the second file.
22. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
23. Write a GUI program to implement CRUD operation on Student record. (rollno, name, percentage)

    [Use data set of your choice from Open Data Portal (https://data.gov.in/) for the following exercises]
24. Practical based on NumPy ndarray
25. Practical based on Pandas Data Structures
26. Practical based on Data Loading, Storage and File Formats
27. Practical based on Interacting with Web APIs
28. Simulate continuous time elementary signals.
29. Simulate basic operations on continuous time elementary signals.
30. Simulate discrete time elementary sequences.
31. Simulate basic operations on discrete time elementary sequences.
32. Simulate sampling on continuous time signals and generate frequency spectrums of signal before and after sampling.
33. Simulate reconstruction of continuous time signals from discrete time sequences, observe effect of sampling rate changes and aliasing in frequency domain spectrums.

34. Simulate a program to analyze discrete time LTI System.
35. Find poles, zeros and gain from a given transfer function and plot it on Z-plane.
36. Simulate the Fourier series representation of a continuous time periodic signal.
37. Simulate frequency domain analysis of discrete time sequences.
38. Simulate correlation and convolution operation discrete time sequences.

Supplementary Resources:

1. https://docs.python.org/3/tutorial/
2. https://www.learnpython.org/
3. https://nptel.ac.in/courses/106/106/106106182/