FACULTY OF ENGINEERING & TECHNOLOGY

Effective from Academic Batch: 2020-21

Programme: **Bachelor of Engineering (Information Technology)**

Semester: **V**

Course Code: **102044504**

Course Title: **PROGRAMMING WITH PYTHON**

Course Group: **Professional Core Course**

Course Objectives: Python Programming is used massively for Artificial Intelligent, Data Science, Web Development and many more. This course is designed with the intention to equip students with the knowledge of Python Programming which intern will motivate them to explore the trending domains of IT. By the end of the course, students will have gained a fundamental understanding of programming in Python by creating a variety of projects using python.

Teaching & Examination Scheme:

Contact hours per week			Course Credits	Examination Marks (Maximum / Passing)				
				Theory		J/V/P*		Total
Lecture	Tutorial	Practical		Internal	External	Internal	External	
3	0	2	4	40 / 14	20 / 07	60 / 21	30 / 10	150 / 52

* **J**: Jury; **V**: Viva; **P**: Practical

Detailed Syllabus:

Sr.	Contents	Hours
1	**Introduction to python and Language fundamentals:** Keywords, Identifiers, Variables, Fundamental data types, working with operators, Taking input and output from the users, Python flow controls and looping, Transfer statements: break and continue.	4
2	**Python data structures and functions:** Working with string and its methods, Python data structures; List, Tuple, Set and Dictionary, Need of function, Types of function, Different types of argument, Types of variables: local and global, Introduction to recursive functions, Working with anonymous functions: Lambda, Map, Filter, Reduce, How to write and use module and its advantages, Introduction to python packages.	7

3	**Object-Oriented Programming and Multi-threading:** Creating class and objects, Introduction to Encapsulation, Abstraction, Polymorphism and Inheritance, Static and instance methods, Constructor, Access modifiers, Getters, setters and deleter behavior through function and decorators, Generator function and iterators, Method overloading, Method overriding, Operator overloading, Interfaces, Abstract method, and abstract class, Working with threads, Introduction to multithreading.	**8**
4	**Exception handling:** Syntax error Vs. Run time error, Introduction to exception handling, Control flow in try-except statement, Try-except-else statement, Try- except-else-Finally, Raising exceptions, Custom exceptions, Basics of Assertion.	**5**
5	**File handling:** Working with files (.txt and .csv) and various file modes, read and write operations with and without with statement, File traversal through tell and seek method, Pickling and unpickling objects into files, Zip and unzip a file.	**5**
6	**Regular expressions and Database:** **Regular expression library - RE and its functions, Web scraping, Connection to MySQL database, Create table, Insert, Select, Update, Where, Order by, Delete, Drop, Limit, Join.**	**7**
7	**GUI design using Tkinter:** **Introduction to Tkinter, Working with different widgets: Entry Widget, Text Widget, Radio Button, Check Button, List Boxes, Menus, and Combo Box.**	**4**
	Total	**40**

List of Practicals / Tutorials:

1	Introduction to various IDE for programming in python.
2	Data Types, Operators and conditional statements: (a) Write a program to print "Hello World". (b) Write a program to find the maximum number out of 3 numbers. (c) Write a program to swap the values of two variables without using temporary variable. (d) Write a program to implement calculator. Write a Python program to calculate the sum of the digits in an integer.
3	Control Structures and Functions: a) Write a program for find factorial of a given number using iterative and recursive function. b) Write a program which takes a sentence from user and calculates number of digits, letters, uppercase letters, lowercase letter and spaces in sentence. c) Write a program to find GCD of two numbers. Write a program to convert Decimal to hex, octal and binary.
4	Working with Data structure - Part I: a) Write a program to remove duplicates from list. b) Write a program to find frequency of elements of list. c) Write a program to sort given list. d) Write a program for matrix addition and matrix multiplication using list. e) Write a program to generate Pascal's triangle using list. f) Write a program to find minimum and maximum value in list of tuples. Write a program to remove an element from tuple.
5	Working with Data structure - Part II: a) Write a program to map 2 lists into a dictionary. b) Write a program to invert keys and values of dictionary. c) Write a program to generate dictionary of frequency of alphabets of given string. d) Write a Python program to sum all the items in a dictionary. e) Write a script to concatenate given dictionaries f) Create a dictionary where keys are name of students and values are another dictionary containing semester, age and cpi of that student. • Print all the names of students. • Print only names of students Print the name of student having highest CPI

6	Object-oriented programming: a) Create a class account with necessary methods – showBalance, withdraw, deposit and transfer. Also, implement necessary getters and setters. Instantiate 2 objects of that class and show different method calls using them. b) Declare a class Person having name as member. Derive two classes a. Businessman - having income and number of people involved in his business as members. b. Employee - having income as a member. c. Create objects of both the classes and compare the income and print the name of one having greater income. c) Create a class clock having 3 init parameters – hours, minutes and seconds. Write following methods: a. setClock – to set the time b. displayTime – to show current time c. tick – increase the time by one second Create a base class A with 2 children B and C. class D having 2 parents B and C. Create a method named "Call" in every class. Use super in a way that a single call to method of class D execute "Call" method of every class.

7	Exception handling and assertions: a) Write a program that opens a file and writes data to it. Handle exceptions that can be generated during the I/O operations. b) Write a program that prompts the user to enter a number. If the number is positive or zero print it, otherwise raise an exception. A message "Code execution completed" should be displayed in both the cases, at the end of the execution. c) Write a program to implement an interactive calculator. User input is expected to be a formula that consist of a number, an operator (at least + and -), and another number, separated by white space (e.g. 1 + 1). Split user input using str.split(), and check whether the resulting list is valid: a. If the input does not consist of 3 elements, raise a FormulaError, which is a custom Exception. b. Try to convert the first and third input to a float (like so: float_value = float(str_value)). Catch any ValueError that occurs, and instead raise a FormulaError. c. If the second input is not '+' or '-', again raise a FormulaError If the input is valid, perform the calculation and print out the result. The user is then prompted to provide new input, and so on, until the user enters quit. d) Handle possible exceptions in the practical 6_a. e) Write a program to find reciprocal of the elements of list [12,0, "a",20,"hi"] with Exception handling. f) Write a program that takes percentage discount on product as input and returns the discounted price of product. Make sure that discount money cannot be less than zero and greater than original price of product using assertion. g) Write a program that calculates division of two vectors and returns the result vector using raise keyword and exception handling.
8	File management: a) Write a program to read n integers from the keyboard and store them into a file total.txt file, separate odd and even numbers and store them in odd.txt and even.txt file. Display the content of all three files. Write a program to copy content of one JSON file into another JSON file.

9	Regular Expressions: a) Write a Python program that matches a string that has an a followed by one or more b's. b) Write a Python program to find sequences of lowercase letters joined with a underscore. c) Write a Python program to find all five characters long word in a string. d) Write a regular expression for checking if given email address is in correct form or not. Correct form is: usename @ domain prefix followed by top level domain. (Top level domains can be – com/edu/org. Username can contain +/./_/-) e) Write a code that takes email id as input. Extract top level domain name and print as output. f) A website requires the users to input username and password to register. Write a program to check the validity of password input by users. Following are the criteria for checking the password: • At least 1 letter between [a-z] • At least 1 number between [0-9] • 1. At least 1 letter between [A-Z] • At least 1 character from [$#@] • Minimum length of transaction password: 6 Maximum length of transaction password: 12
10	Database operations: a) Write a Python program to create a table named student with id, name and spi data. b) Insert data in student table. c) Display the list of students having spi greater than 8. d) Create a table Course with id, name and student id. Display the details of students who are registered for a specific course.
11	Develop programs to learn GUI programming using Tkinter.

Reference Books:

1	John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India
2	R Nageswara Rao, Core Python Programming, 2nd Edition, Dreamtech Press
3	Hans Petter Langtangen, A Primer on Scientific Programming with Python, Springer
4	Vernon L. Ceder, The Quick Python Book, Second Edition, By Manning Publications
5	Wesley J. Chun. "Core Python Programming - Second Edition", Prentice Hall
6	Daniel Y. Chen, Pandas for everyone: Python Data Analysis, Pearson

Supplementary learning Material:	
1	Coursera: Python for Everybody Specialization (https://www.coursera.org/specializations/python)
2	edX: CS50's Introduction to Programming with Python
3	https://python.fossee.in/spoken-tutorials/

4	https://pythonguides.com/numpy/
5	https://pythonguides.com/pandas-in-python/

Pedagogy:

- Direct classroom teaching
- Audio Visual presentations/demonstrations
- Assignments/Quiz
- Continuous assessment
- Interactive methods
- Seminar/Poster Presentation
- Industrial/ Field visits
- Course Projects

Suggested Specification table with Marks (Theory) (Revised Bloom's Taxonomy):

Distribution of Theory Marks in %						**R**: Remembering; **U**: Understanding; **A**: Applying; **N**: Analyzing; **E**: Evaluating; **C**: Creating
R	**U**	**A**	**N**	**E**	**C**	
15%	25%	25%	15%	20%	-	

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes (CO):

Sr.	Course Outcome Statements	%weightage
CO-1	Write, Test and Debug Python Programs	**45**
CO-2	Use functions and represent Compound data using Lists, Tuples and Dictionaries	**15**
CO-3	Text filtering with regular expressions in Python.	**25**
CO-4	Data manipulation and data visualization in Python.	**15**
CO-5	Create GUI applications in Python.	

Curriculum Revision:	
Version:	1.0
Drafted on (Month-Year):	June-2020
Last Reviewed on (Month-Year):	-
Next Review on (Month-Year):	June-2025