COURSE TITLE	**PYTHON PROGRAMMING & APPLICATIONS**
COURSE CODE	**01EC0115**
COURSE CREDITS	**2**

Objective:

1 To equip students with hands-on programming skills in Python, covering fundamental concepts, data handling, and visualization, and applying them to simulate and prototype basic systems in Electronics and Communication Engineering.

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

1 Apply Python programming constructs, including functions, data structures, and OOP, to solve problems involving strings, files, and exceptions. (Bloom's Level: Apply)

2 Analyze sensor and engineering data using NumPy, Pandas, and Matplotlib for effective interpretation. (Bloom's Level: Analyze)

3 Evaluate the efficiency, readability, and scalability of Python programs to select optimal solutions for engineering problems. (Bloom's Level: Evaluate)

4 Design user-defined classes and develop Python-based prototypes for modeling real-world entities and ECE applications like signal processing and IoT. (Bloom's Level: Create)

Pre-requisite of course: Basic knowledge of Programming.

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
0	0	4	0	30	20	25	25

Contents : Unit	Topics	Contact Hours
1	**Python Basics and Programming Environment** Setting up Python using IDEs like IDLE, VS Code, or Jupyter, etc.; Introduction to Python syntax and structure; , data and expressions: literals; variables and identifiers; operators; expressions and data types; Boolean expressions	2
2	**Experiment based on Python basics** 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 access a substring from a given string. , 4) Write a Python script to print the current date in the format "Sun August 01 10:10:10 IST 2025"., 5) Write a Python program to convert temperature to and from Celsius to Fahrenheit., 6) Write a simple calculator, unit converter, and basic I/O script.	2

DR. RAJENDRAKUMAR DAHYALAL PATEL
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Contents : Unit	Topics	Contact Hours
3	**Control Structures** Conditional statements (if, elif, else), Loops (for, while, break, continue)	1
4	**Experiment based on Control structure** 1) Develop programs to understand the control structures of Python. 2) Write a Python program to find the largest of three numbers., 3) Write a Python program to print prime numbers less than a given number. 4) Write a Python program that accepts lengths of three sides of a triangle and checks for right-angled triangle., 5) Write a prime number check, pattern printing, and menu-driven program. 6) Write a program for evaluating a Boolean function.	3
5	**Functions and Recursion** Defining functions, Calling Value Returning Functions; Calling Non-value Returning Functions,, Parameter Passing; Keyword and Default Arguments in Python; Recursion basics and use-cases	2
6	**Experiment based on Function and Recursion** 1) Write a Python program to find the factorial of a number using recursion. 2) Write a Python program to define a module to find Fibonacci Numbers and import the module to another program., 3) Write a Python program to define a module and import a specific function from that module. 4) Write a program for factorial, Fibonacci, power computation, and number system converters.	2
7	**Python Data Structures – Lists and Tuples** Statements, Lists: indexing, slicing, methods; Tuples: immutability, usage	1
8	**Experiment based on Python Data Structures – Lists and Tuples** 1) Write a Python program to create, append, and remove items from a list., 2) Write a program to demonstrate working with tuples in Python., 3) Write a program for list manipulation (marks, matrix) and tuple-based sensor input simulation.	3
9	**Python Data Structures – Sets and Dictionaries** Set operations, applications Dictionaries: key-value pairs, methods.	1
10	**Experiment based on Python Data Structures – Sets and Dictionaries** 1) Write a program to demonstrate working with dictionaries in Python., 2) Write a word frequency counter, student record system, and unique sensor data filter.	3
11	**String Handling** String operations, formatting, slicing, searching	1
12	**Experiment based on String Handling** 1) Create, concatenate and print strings, and access substrings., 2) Write a palindrome check, string analyzer, and encrypted communication simulation.	3
13	**File Handling and CSV I/O** Reading and writing to text and CSV files, Parsing structured data	1

DR. RAJENDRAKUMAR DAHYALAL PATEL
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Contents : Unit	Topics	Contact Hours
14	**Experiment based on File handling and CSV I/O** 1) Write a script named copyfile.py to copy file contents., 2) Write a program to read a text file and print unique words alphabetically., 3) Simulate a data logger, generate a CSV-based report, and read configuration files.	3
15	**Object-Oriented Programming** Classes, objects, methods, inheritance; Constructor and encapsulation	1
16	**Experiment based on OOP** 1) Call data member and function using classes and objects., 2) Read 3 subject marks and display pass or fail using class and object., 3) Create a class for sensors, inheritance for analog/digital sensors, and perform object comparison.	3
17	**Exception Handling and Debugging** Basic understanding of Exceptions, Try-except blocks, user-defined exceptions; Debugging with breakpoints/logging	1
18	**Experiment based on Exception handling and Debugging** Validate PAN card number and Email ID., Create a robust calculator with input validation and error-logging system.	3
19	**Introduction to Libraries** Motivation for using python for data analysis, introduction of Jupyter Notebook,, Overview of NumPy, SciPy; NumPy arrays, , operations, slicing; Plotting with Matplotlib: line, bar, scatter	3
20	**Experiment based on Introduction to libraries** Use datetime library to display current date and time., Perform practical based on interacting with Web APIs., Perform sensor trend analysis, Fourier Transform basics, and signal/temperature log analysis.	3
21	**Introduction to Pandas and Data Frames** Pandas library for data manipulation and analysis, Creating and manipulating Data Frames	1
22	**Experiment based on Pandas and Data Frames** 1) Perform practical based on NumPy and array. 2) Perform practical based on Pandas Data Structures., 3) Perform practical based on Data Loading, Storage, and File Formats. 4) Develop a GUI calculator using pandas for I/O., 5) Carry out Spectrum analysis using Fourier Transform in Python.	3
23	**GUI Programming with Tkinter (or Streamlit for Web UI)** GUI basics: buttons, labels, text fields, Event handling	1
24	**Experiment based on GUI programming with Tkinter** 1) Create a GUI program for Tic-tac-toe. 2) Create a GUI program to implement CRUD operation on Student record. , 3) Develop a GUI calculator., 4) Implement a form-based data entry system for sensor configuration.	3

DR. RAJENDRAKUMAR DAHYALAL PATEL
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Contents : Unit	Topics	Contact Hours
25	**ECE Applications Using Python** Signal generation and visualization (sinusoidal, square, triangle), Basic DSP (Fourier Transform, filtering), Logic gate simulation, Simulated serial communication (e.g., UART using pySerial), Mocking Arduino/Raspberry Pi sensor input via CSV/logs	1
26	**Experiment** 1) Analyze sensor trends, perform Fourier Transform, analyze temperature logs and signal stats., 2) Plot various electronic signals using NumPy & Matplotlib., 3) Carry out Spectrum analysis using Fourier Transform in Python., 4) Simulate signal shapes using Python.	5
Total Hours		**56**

Textbook :

1 Learning Python, Mark Lutz, O'Reilly, 4th Edition, 2009

2 Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2nd Edition, 2009

References:

1 Programming Python, Programming Python, Mark Lutz, O'Reilly 4th Edition, 2010

2 Python 3 for Absolute Beginners, Python 3 for Absolute Beginners, Tim Hall, J-P Stacey, Apress, 2009

3 Introduction to Computer Science Using Python: A Computational Problem-Solving Focus, Introduction to Computer Science Using Python: A Computational Problem-Solving Focus, C. Dierbach, Wiley, 2015

4 Let Us Python, Let Us Python, Yashavant Kanetkar, BPB Publications, 2019

5 Think Python: How to Think Like a Computer Scientist, Think Python: How to Think Like a Computer Scientist, Allen B. Downey, O'Reilly, 2015

6 Python: The Complete Reference, Python: The Complete Reference, Martin C. Brown, McGraw-Hill Education, 2001

7 Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython, Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython, Wes McKinney, O'Reilly Media, 2017

8 Doing Data Science: Straight Talk from the Frontline, Doing Data Science: Straight Talk from the Frontline, Cathy O'Neil, Rachel Schutt, O'Reilly Media, 2013

9 Python for Signal Processing, Python for Signal Processing, José Unpingco, Springer, 2014

Suggested Theory Distribution:

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

Distribution of Theory for course delivery and evaluation

DR. RAJENDRAKUMAR DAHYALAL PATEL
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Remember / Knowledge	Understand	Apply	Analyze	Evaluate	Higher order Thinking / Creative
5.00	15.00	40.00	15.00	10.00	15.00

Instructional Method:

1. PowerPoint Presentations and Visual Aids
2. Hands-On Demonstrations
3. Step-by-Step Guided Lab Work
4. Code-Along Sessions
5. Problem-Based Learning (PBL)
6. Demonstration-Based Learning
7. Peer Learning
8. Live Debugging Sessions
9. Quiz & Reflection
10. Day-long coding / Hackathon

Supplementary Resources:

1. NPTEL (SWAYAM), Course: The Joy of Computing using Python by Prof. NPTEL Faculty, IIT Ropar Link: https://swayam.gov.in/nd1_noc19_cs42/preview , Highlights: Tailored for beginners with problem-solving orientation, Includes topics like file handling, libraries, and basic data structures, Free enrollment, with optional certification.
2. Coursera, Course: Python for Everybody by Dr. Charles Severance (University of Michigan), Link: https://www.coursera.org/specializations/python , Highlights: Comprehensive course for absolute beginners to intermediate learners, Focuses on programming fundamentals with real-world examples, Free access (audit mode), paid certification available.
3. Udemy, Course: Complete Python Bootcamp: Go from zero to hero in Python 3 by Jose Portilla, Link: https://www.udemy.com/course/complete-python-bootcamp/ , Highlights: Hands-on approach with exercises and projects, Covers basics to OOP, file I/O, and modules—ideal for ECE students exploring automation or data analysis, Lifetime access and affordable pricing.
4. MIT OpenCourseWare, Course: 6.0001 – Introduction to Computer Science and Programming in Python Link: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/ , Highlights: Rigorous, theory-backed course from MIT, Useful for students aiming for deeper understanding of programming in scientific computing contexts.
5. The Python Tutorial : https://docs.python.org/3/tutorial/
6. Getting Started with Python : https://www.learnpython.org/