FACULTY OF ENGINEERING AND TECHNOLOGY

Effective from Academic Batch: 2022-23

Programme: **M.TECH. ARTIFICIAL INTELLIGENCE**

Semester: **I**

Course Code: **202340104**

Course Title: **Python Programming**

Course Group: **Programme Elective-I**

Course Objectives: (1) To teach students to program in Python in a practical and hands-on manner using the industry standard methods, tools and technologies. (2) To improve students' algorithmic thinking and problem solving capabilities so that they can write code that actually works and produces the desired functional results.

Teaching & Examination Scheme:

Contact hours per week			Course Credits	Examination Marks (Maximum / Passing)				
Lecture	Tutorial 1	Practical 1		Theory		J/V/P*		Total
				Internal	External	Internal	External	
3	0	2	4	50/20	50/20	25/10	25/10	150/60

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

Detailed Syllabus:

Sr.	Contents	Hours
1	**Introduction to Python:** The basic elements of python, Branching Programs, Control Structures, Strings and Input, Iteration	**03**
2	**Functions, Scoping and Abstraction:** Functions and scoping, Specifications, Recursion, Global variables, Modules, Files, System Functions and Parameters, Decorators, Generators	**05**
3	**Structured Types, Mutability and Higher-Order Functions:** Strings, Tuples, Lists and Dictionaries, Lists and Mutability , Functions as Objects	**04**
4	**Testing, Debugging, Exceptions and Assertions**: Types of testing – Black-box and Glass-box, Debugging, Handling Exceptions, Assertions	**03**
5	**Classes and Object-Oriented Programming:** Abstract Data Types and Classes, Inheritance, Encapsulation and Information Hiding, Polymorphism	**04**
6	**Simple Algorithms and Data structures:** Search Algorithms, Sorting Algorithms, Hash Tables	**06**

7	**Advanced Topics I:** Regular Expressions – REs and Python, Networking and Multithreaded Programming – Sockets, Threads and Processes, Chat Application	**07**
8	**Advanced Topics II:** Plotting using PyLab, Security – Encryption and Decryption, Classical Ciphers, GUI Programming –Tkinter and Python	**08**

List of Practical's / Tutorials:

1	Develop programs to understand the control structures of python
2	Develop programs to learn different types of structures (list, dictionary, tuples) in python
3	Develop programs to learn concept of functions scoping, recursion and list mutability.
4	Develop programs to understand working of exception handling and assertions.
5	Develop programs for data structure algorithms using python – searching, sorting and hash tables.
6	Develop programs to learn regular expressions using python.
7	Develop chat room application using multithreading.
8	Learn to plot different types of graphs using PyPlot
9	Develop programs to understand file handling using python
10	Develop programs to implement classical ciphers.
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", dreamtech
3	Wesley J. Chun. "Core Python Programming - Second Edition", Prentice Hall
4	Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, "Data Structures and Algorithms in Pyhon", Wiley
5	Kenneth A. Lambert, "Fundamentals of Python – First Programs", CENGAGE Publication
6	Luke Sneeringer, "Professional Python", Wrox

Supplementary learning Material:	
1	Lecture Note
2	https://nptel.ac.in/courses/106/106/106106145/
3	https://nptel.ac.in/courses/106/106/106106212/

Pedagogy:

- Direct classroom teaching
- Audio Visual presentations/demonstrations
- Assignments/Quiz
- Continuous assessment
- Interactive methods
- Seminar/Poster Presentation

- Industrial/ Field visits
- Course Projects

Internal Evaluation:

The internal evaluation comprised of written exam (40% weightage) along with combination of various components such as Certification courses, Assignments, Mini Project, Simulation, Model making, Case study, Group activity, Seminar, Poster Presentation, Unit test, Quiz, Class Participation, Attendance, Achievements etc. where individual component weightage should not exceed 20%.

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**	
10%	40%	30%	10%	--	10%	

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	To understand various data structures available in Python programming language and apply them in solving computational problems.	**30**
CO-2	To do testing and debugging of code written in Python.	**10**
CO-3	To draw various kinds of plots using PyLab.	**15**
CO-4	To do text filtering with regular expressions in Python	**15**
CO-5	To create socket applications in Python	**10**
CO-6	To process numerical data using various modules available	**20**

Curriculum Revision:	
Version:	2.0
Drafted on (Month-Year):	June-2022
Last Reviewed on (Month-Year):	-
Next Review on (Month-Year):	June-2025