

## PYTHON PROGRAMMING

I Semester: Common for all branches								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
ACSC01	Foundation	L	T	P	C	CIA	SEE	Total
		3	0	0	3	30	70	100
<b>Contact Classes: 45</b>		<b>Tutorial Classes: Nil</b>		<b>Practical Classes: Nil</b>		<b>Total Classes: 45</b>		
<b>Prerequisites: There are no prerequisites to take this course.</b>								
<b>I. COURSE OVERVIEW:</b>								
<p>This course introduces students to writing computer programs. This course presents the principles of structured programming using the Python language, one of the most increasingly preferred languages for programming today. Because of its ease of use, it is ideal as a first programming language and runs on both the PC and Macintosh platforms. However, the knowledge gained in the course can be applied later to other languages such as C and Java. The course uses Python Notebook to afford a more interactive experience. Topics include fundamentals of computer programming in Python, object-oriented programming and graphical user interfaces.</p>								
<b>II. COURSE OBJECTIVES:</b>								
<b>The students will try to learn:</b>								
<p>I. Acquire programming skills in core Python.            II. Acquire Object-oriented programming skills in Python.            III. Develop the skill of designing graphical-user interfaces (GUI) in Python.            IV. Develop the ability to write database applications in Python.            V. Acquire Python programming skills to move into specific branches - Internet of Things (IoT), Data Science, Machine Learning (ML), Artificial Intelligence (AI) etc.</p>								
<b>III. COURSE OUTCOMES:</b>								
<b>After successful completion of the course, students should be able to:</b>								
CO 1	Demonstrate the basic concepts of python programming with the help of data types, operators, expressions, and console input/output.						Understand	
CO 2	Make use of control statements for altering the sequential execution of programs in solving problems.						Apply	
CO 3	Demonstrate operations on built-in container data types (list, tuple, set, dictionary) and strings.						Understand	
CO 4	Illustrate operations and applications on strings with the help of built in functions.						Understand	
CO 5	Solve the problems by using modular programming concepts through functions.						Apply	
CO 6	Identify object oriented programming constructs for developing large, modular and reusable real-time programs.						Apply	
<b>IV. SYLLABUS:</b>								
<b>MODULE – I: INTRODUCTION TO PYTHON (09)</b>								
Introduction to Python: Features of Python, History and Future of Python, Working with Python – interactive and script mode, Identifiers and Keywords, Comments, Indentation and Multi-lining, Data types – built-in data types, Operators and Expressions, Console Input/Output, Formatted printing, Built-in Functions, Library Functions.								
<b>MODULE – II: DECISION CONTROL STATEMENTS (09)</b>								
Selection/Conditional Branching Statements: if, if-else, nested if, if-elif-else statement(s), Basic Loop Structures/ Iterative Statements – while and for loop, Nested loops, break and continue statement, pass Statement, else Statement used with loops.								
<b>MODULE – III: CONTAINER DATA TYPES (09)</b>								
Lists: Accessing List elements, List operations, List methods, List comprehension; Tuples: Accessing Tuple elements, Tuple operations, Tuple methods, Tuple comprehension, Conversion of List comprehension to Tuple, Iterators and Iterables, zip() function.								

Sets: Accessing Set elements, Set operations, Set functions, Set comprehension; Dictionaries: Accessing Dictionary elements, Dictionary operations, Dictionary Functions, Nested Dictionary, Dictionary comprehension.

#### **MODULE – IV: STRINGS AND FUNCTIONS (09)**

Strings: Accessing String elements, String properties, String operations.

Functions: Communicating with functions, Variable Scope and lifetime, return statement, Types of arguments, Lambda functions, Recursive functions.

#### **MODULE – V: CLASSES AND OBJECTS (09)**

Classes and Objects – Defining Classes, Creating Objects, Data Abstraction and Hiding through Classes, Class Method and self Argument, Class variables and Object variables, `__init__` and `__del__` method, Public and private data members, Built-in Class Attributes, Garbage Collection. OOPs Features: Abstraction, Encapsulation, Inheritance, and Polymorphism.

#### **V. TEXT BOOKS:**

1. Reema Thareja, “Python Programming - Using Problem Solving Approach”, Oxford Press, 1<sup>st</sup> Edition, 2017.
2. Dusty Philips, “Python 3 Object Oriented Programming”, PACKT Publishing, 2<sup>nd</sup> Edition, 2015.

#### **VI. REFERENCE BOOKS:**

1. Yashavant Kanetkar, Aditya Kanetkar, “Let Us Python”, BPB Publications, 2<sup>nd</sup> Edition, 2019.
2. Martin C. Brown, “Python: The Complete Reference”, Mc. Graw Hill, Indian Edition, 2018.
3. Michael H. Goldwasser, David Letscher, “Object Oriented Programming in Python”, Prentice Hall, 1<sup>st</sup> Edition, 2007.
4. Taneja Sheetal, Kumar Naveen, “Python Programming – A Modular Approach”, Pearson, 1<sup>st</sup> Edition, 2017.  
R Nageswar Rao, “Core Python Programming”, Dreamtech Press, 2018.

#### **VII. WEB REFERENCES:**

1. <https://realPython.com/Python3-object-oriented-programming/>
2. <https://Python.swaroopch.com/oop.html>
3. [https://Python-textbok.readthedocs.io/en/1.0/Object\\_Oriented\\_Programming.html](https://Python-textbok.readthedocs.io/en/1.0/Object_Oriented_Programming.html)
4. <https://www.programiz.com/Python-programming/>