

II Year - I Semester

L	T	P	C
4	0	0	3

## PYTHON PROGRAMMING

### OBJECTIVES:

- Introduction to Scripting Language
- Exposure to various problems solving approaches of computer science

### UNIT – I:

**Introduction:**History of Python, Need of Python Programming, Applications Basics of Python Programming Using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input-Output, Indentation.

### UNIT – II:

**Types, Operators and Expressions:** Types - Integers, Strings, Booleans; Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations Control Flow- if, if-elif-else, for, while, break, continue, pass

### UNIT – III:

**Data Structures** Lists - Operations, Slicing, Methods; Tuples, Sets, Dictionaries, Sequences. Comprehensions.

### UNIT – IV:

**Functions** - Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Anonymous Functions, Fruitful Functions(Function Returning Values), Scope of the Variables in a Function - Global and Local Variables.

**Modules:** Creating modules, import statement, from. Import statement, name spacing,

**Python packages,** Introduction to PIP, Installing Packages via PIP, Using Python Packages

### UNIT – V:

**Object Oriented Programming OOP in Python:** Classes, 'self variable', Methods, Constructor Method, Inheritance, Overriding Methods, Datahiding,

**Error and Exceptions:** Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User Defined Exceptions

**UNIT – VI:**

**Brief Tour of the Standard Library** - Operating System Interface - String Pattern Matching, Mathematics, Internet Access, Dates and Times, Data Compression, Multithreading, GUI Programming, Turtle Graphics

**Testing:** Why testing is required ?, Basic concepts of testing, Unit testing in Python, Writing Test cases, Running Tests.

**OUTCOMES:**

- Making Software easily right out of the box.
- Experience with an interpreted Language.
- To build software for real needs.
- Prior Introduction to testing software

**TEXT BOOKS**

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
2. Learning Python, Mark Lutz, Orielly

**Reference Books:**

1. Think Python, Allen Downey, Green Tea Press
2. Core Python Programming, W.Chun, Pearson.
3. Introduction to Python, Kenneth A. Lambert, Cengage