

## PYTHON PROGRAMMING

**Credits: 4**

**Semester:III**

**Subject Code: DS20301**

**No. of Lecture Hours : 60**

**Objective:** To offer an easy syntax compared to Perl and the Unix/Linux “shell languages” and it is easier to learn and maintain.

**Outcomes:** Students will be able to

**CO1:** Implement the structure and components of a Python program.

**CO2:** Choose appropriate data structures.

**CO3:** Interpret how to write classes and create objects.

**CO4:** Explain how to use Inheritance and Threads.

**CO5:** Create basic GUI programs and Database Connectivity

### UNIT –I

**12hrs**

#### **1. Introduction to Python:**

**3**

Python, Features of Python, Execution of a Python Program  
Viewing the Byte Code, Flavors of Python, Python Virtual Machine, Frozen Binaries,  
Memory Management in Python, Garbage Collection in Python,  
Comparisons between C and Python, Comparisons between Java and Python.

#### **2. Writing Our First Python Program:**

**3**

Writing Our First Python  
Program, Executing a Python Program - Using Python’s Command Line Window  
Using Python’s IDLE Graphics Window  
Running from System Prompt, Getting Help in Python,  
Getting Python Documentation Help.

#### **3. Data types in Python:**

**3**

Comments in Python, Doc Strings, How Python Sees Variables, Data types in Python  
Built-in Data Types, Bool Data type.  
Sequences in Python, Sets, Literals in Python,  
Determining the Data type of a Variable What about Characters,  
User-defined Data types, Constants in Python, Identifiers and  
Reserved words, Naming Conventions in Python.

#### **4. Operators in Python:**

**3**

Arithmetic Operators, Assignment Operators,Unary Minus Operator,  
Relational Operators, Logical Operators, Boolean Operators, Bitwise Operators,  
Membership Operators, Identity Operators.  
Operator Precedence and Associativity, Mathematical Functions

Using IDLE Window, Using Command Line Window.

**Input and Output:** Output statements, Input Statements,  
Command Line Arguments

## UNIT- II

12Hrs

1. **Control Statements:** The if Statement, A Word on Indentation, The if ... else Statement, The if .. elif .. else Statement, The while Loop, The for Loop, Infinite Loops, Nested Loops, The else Suite, The break Statement, The continue Statement, The pass Statement, The assert Statement, The return Statement. 3
2. **Arrays in Python:** 3  
Array, Advantages of Arrays, Creating an Array, Importing the Array Module, Indexing and Slicing on Arrays, Processing the Arrays, Types of Arrays. Working with Arrays using Numpy.  
Creating arrays with array(), linspace(), logspace(), arange(), zeros() and ones() functions
3. **Strings and Characters:** 3  
Creating Strings, Length of a String, Indexing in Strings, Slicing the Strings, Repeating the Strings, Concatenation of Strings, Checking Membership, Comparing Strings, Removing Spaces from a String, Finding Sub Strings, Counting Substrings in a String, Strings are Immutable, Replacing a String with another String, Splitting and Joining Strings, Changing Case of a String, Checking Starting and Ending of a String, String Testing Methods, Formatting the Strings, Working with Characters, Sorting Strings, Searching in the Strings, Finding Number of Characters and Words, Inserting Sub String into a String,  
**Regular Expressions**, Sequence Characters, Quantifiers and Special Characters in Regular Expressions.
4. **Functions:** 3  
Difference between a Function and a Method, Defining a Function, Calling a Function, Returning Results from a Function, Returning Multiple Values from a Function, Functions are First Class Objects, Pass by Object Reference, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Variable Length Arguments, Local and Global Variables, The Global Keyword, Passing a Group of Elements to a Function, Recursive Functions, Anonymous Functions or Lambdas Function Decorators, Generators, Structured Programming.  
Creating our Own Modules in Python, The Special Variable name.

## UNIT -III

12hrs

- 1. Lists and Tuples:** 3  
List, Creating Lists using range() Function, Updating the Elements of a List, Concatenation of Two Lists, Repetition of Lists, Membership in Lists, Aliasing and Cloning Lists, Methods to Process Lists  
Finding Biggest and Smallest Elements in a List, Sorting the List Elements, Number of Occurrences of an Element in the List, Finding Common Elements in Two Lists, Storing Different Types of Data in a List, Nested Lists, Nested Lists as Matrices, List Comprehensions.  
Tuples: Creating Tuples,  
Accessing the Tuple Elements, Basic Operations on Tuples, Functions to Process Tuples, Nested Tuples, Inserting Elements in a Tuple, Modifying Elements of a Tuple, Deleting Elements from a Tuple.
- 2. Dictionaries:** 3  
Operations on Dictionaries, Dictionary Methods, Using for Loop with Dictionaries, Sorting the Elements of a Dictionary using Lambdas, Converting Lists into Dictionary, Converting Strings into Dictionary, Passing Dictionaries to Functions, Ordered Dictionaries
- 3. Introduction to OOPs:** 3  
Problem in Procedure Oriented Approach,  
Specialty of Python Language, Features of Object Oriented Programming System (OOPs)- Classes and Objects, Encapsulation, Abstraction, Inheritance, Polymorphism.
- 4. Classes and Objects:** 3  
Creating a Class, the Self Variable, Constructor,  
Types of Variables, Namespaces, Types of Methods- Instance Methods, Class Methods, Static Methods, Passing Members of One Class to another Class, Inner Classes

## UNIT-IV

12hrs

- 1. Inheritance and Polymorphism:** 3  
Constructors in Inheritance, Overriding Super Class Constructors and Methods, The super() Method, Types of Inheritance, Single Inheritance, Multiple Inheritance, Method Resolution Order(), Polymorphism, Duck Typing  
Philosophy of Python, Operator Overloading, Method Overloading, Method Overriding.
- 2. Exceptions:** 3  
Exceptions in Python Program, Exceptions, Exception Handling, Types of Exceptions, The Except Block, The assert Statement, User-Defined Exceptions, Logging the Exceptions.

- 3. Threads:** 3
- Single Tasking, Multitasking, Difference between a Process and a Thread, Concurrent Programming and GIL, Uses of Threads
- Creating Threads in Python- Creating a Thread without using a class, Creating a Thread by Creating a Sub Class to Thread Class, Creating a Thread without Creating a Sub Class to Thread Class, Thread Class Methods – Single Tasking using a Thread, Multitasking using Multiple Threads, Thread Synchronization, -Locks, Semaphores, Deadlock of Threads, Avoiding Deadlocks in a Programs, Communication between Threads using notify(), wait() methods and using Queue, Daemon Threads

**UNIT -V** 12Hrs

- 1. Files in Python:** 3
- Files, Types of Files in Python, Opening a File, Closing a File, Working with Text Files Containing Strings, knowing whether a File Exists or Not, Working with Binary Files, The with Statement, Pickle in Python, The seek() and tell() Methods, Working with Directories.

- 2. Graphical User Interface:** 3
- GUI in Python, The Root Window, Fonts and Colors, Working with Containers, Canvas, Frame, Widgets, Button Widget, Arranging Widgets in the Frame, Label Widget, Message Widget, Text Widget, Scrollbar Widget, Checkbutton Widget, Radiobutton Widget, Entry Widget, Spinbox Widget, Listbox Widget, Menu Widget, Creating Tables

- 3. Python's Database Connectivity:** 3
- DBMS, Advantages of DBMS over Files, Types of Databases used in Python, Working with MySQL Database, Using MySQL from Python, Retrieving All Rows from a Table, Inserting Rows into a Table, Deleting Row from a Table, Updating Rows in a Table, Creating Database Tables through Python