PR-3 PYTHON PROGRAMMING LAB

Total Periods	**60**	**Maximum Marks**	**75 Marks**
Lab. Periods:	**4 Periods /week**	**Term Works**	**25 Marks**
Examination	**3hours**	**End Semester Examination**	**50Marks**

RATIONALE

This course introduces to the students the Python language. Upon completion of this course, the student will be able to write non trivial Python programs dealing with a wide variety of subject matter domains. Topics include language components, the IDLE/IDE environment, control flow constructs, strings, I/O, collections, classes, modules, and regular expressions

LEARNING OUTCOMES

After undergoing the course, the students will be able to:

- Execute Python code in a variety of environments
- Use correct Python syntax in Python programs
- Use the correct Python control flow construct
- Write Python programs using various collection data types
- Write home grown Python functions
- Use many of the standard Python modules such as os, sys, math, and time
- Trap various errors via the Python Exception Handling model
- Use the IO model in Python to read and write disk files
- Create their own classes and use existing Python classes. Understand and use the Object Oriented paradigm in Python programs
- Use the Python Regular Expression capabilities for data verification

DETAILED CONTENTS

1. Introduction
    - Brief History of Python
    - Python Versions
    - Installing Python
    - Environment Variables
    - Executing Python from the Command Line
    - IDLE
    - Editing Python Files
    - Python Documentation
    - Getting Help
    - Dynamic Types
    - Python Reserved Words
    - Naming Conventions

2. Basic Python Syntax
    - ▪ Basic Syntax
    - ▪ Comments
    - ▪ String Values
    - ▪ String Methods
    - ▪ The format Method

- String Operators
- Numeric Data Types
- Conversion Functions
- Simple Output
- Simple Input
- The % Method
- The print Function

3. Language Components
   - Indenting Requirements
   - The if Statement
   - Relational and Logical Operators
   - Bit Wise Operators
   - The while Loop
   - break and continue
   - The for Loop

4. Collections
   - Introduction
   - Lists
   - Tuples
   - Sets
   - Dictionaries
   - Sorting Dictionaries
   - Copying Collections
   - Summary

5. Functions
   - Introduction
   - Defining Your Own Functions
   - Parameters
   - Function Documentation
   - Keyword and Optional Parameters
   - Passing Collections to a Function
   - Variable Number of Arguments
   - Scope
   - Functions - "First Class Citizens"
   - Passing Functions to a Function
   - map
   - filter
   - Mapping Functions in a Dictionary
   - Lambda
   - Inner Functions
   - Closures

6. Modules
   - Modules
   - Standard Modules - sys
   - Standard Modules - math
   - Standard Modules - time
   - The dir Function

7. Exceptions
   - Errors
   - Runtime Errors
   - The Exception Model
   - Exception Hierarchy
   - Handling Multiple Exceptions
   - Raise
   - assert

8. Input and Output
   - Introduction
   - Data Streams
   - Creating Your Own Data Streams
   - Access Modes
   - Writing Data to a File
   - Reading Data From a File
   - Additional File Methods
   - Using Pipes as Data Streams
   - Handling IO Exceptions

9. Classes in Python
   - Classes in Python
   - Principles of Object Orientation
   - Creating Classes
   - Instance Methods
   - File Organization
   - Special Methods
   - Class Variables
   - Inheritance
   - Polymorphism

10. Regular Expressions
   - Introduction
   - Simple Character Matches
   - Special Characters
   - Character Classes
   - Quantifiers
   - The Dot Character
   - Greedy Matches
   - Grouping
   - Matching at Beginning or End
   - Match Objects
   - Substituting
   - Splitting a String
   - Compiling Regular Expressions
   - Flags

LIST OF PRACTICALS

1. Write instructions to perform each of the steps below

   (a) Create a string containing at least five words and store it in a variable.

   (b) Print out the string.

   (c) Convert the string to a list of words using the string split method.

(d) Sort the list into reverse alphabetical order using some of the list methods (you might need to use dir(list) or help(list) to find appropriate methods).

(e) Print out the sorted, reversed list of words.

2. Write a program that determines whether the number is prime.
3. Find all numbers which are multiple of 17, but not the multiple of 5, between 2000 and 2500?
4. Swap two integer numbers using a temporary variable. Repeat the exercise using the code format: a, b = b, a. Verify your results in both the cases.
5. Find the largest of n numbers, using a user defined function largest().
6. Write a function my Reverse() which receives a string as an input and returns the reverse of the string.
7. Check if a given string is palindrome or not.
8. WAP to convert Celsius to Fahrenheit
9. Find the ASCII value of charades
10. WAP for simple calculator

Methodology

The Students shall be taught about the Syntax of Python Language which is similar to other High level languages in the initial 8 to 10 classes. Then after learning the syntax the students shall Write the codes for the Practical Exercise and Test its results in the Lab.

RECOMMENDED BOOKS

Sl.No	**Name of Authors**	**Title of the Book**	**Name of the publisher**
1	C Satyanarayan M, Radhika Mani, B N Jagdesh	Python Programming	University Press
2	Mark Lutz;	Learning Python by Mark Lutz;	Pratham Books, Bangalore
3	Robert Richards	Python Programming For Beginners: A Must Read Introduction to Python Programming	Pratham Books, Bangalore