Python Programming Lab

Course Code	Course Title				Category
	Python Programming Lab				
Contact Hours per Week			CA	FE	Credits
L	T	D/P			
0	0	4	20	30	2

Prerequisite:

- Fundamentals of Programming Languages

Course Objectives:

- To introduce programming concepts using python
- To develop programs using various language components and data structures
- To develop high level applications using functions, classes, and modules
- To handle errors, test and execute python programs

COURSE CONTENT

Unit 1: Introduction to Python Programming Language. :

Introduction to Python Language - Strengths and Weaknesses, IDLE, Dynamic Types, Naming Conventions,

Data Types, Variables, Basic Input-Output Operations, Basic Operators : Standard data types - basic, none, Boolean (true & False), numbers, Variables, Constants, Python identifiers and reserved words, Lines and indentation, multi-line statements and Comments, Input/output with print and input, functions Declaration, Operations on Data such as assignment, arithmetic, relational, logical and bitwise operations, dry run, Simple Input and output

String - Values, Operations, Slices, Operators, Numeric Data Types, Conversions, Built In Functions

—--

Problem Statement 1: (print with attributes)

Use print in Python to produce the following outputs:

a. Programming***Essentials***in...Python
b. Nursery rhyme of your choice (with proper indentation and newline)
c. Assign three string values say “A”, “B”, “C” to the variables and print them as below: HI, A B C

Problem Statement 2: (Operators)

Your task is to read the input numbers from the user [on the same line] and implement basic calculations - Addition, subtraction, division, floor division, multiplication, Exponentiation, Modulus

Problem Statement 3: Write python program for string operations- copy, concatenate, check substring, equal, reverse and length.

—---

Unit 2: Data Collections and Language Component

Introduction - Control Flow and Syntax, Indenting, The if Statement, The while Loop, break and continue, for Loop

Lists - Concept, creating and accessing elements, updating & deleting lists, traversing a List, reverse Built-in List Operators, Concatenation, In Operator, Built-in List functions and methods

Tuples, Sets, Dictionaries, Sorting Dictionaries, Copying Collections.

—---

Problem Statement 4: (Loops)

a. Accept two numbers from the user (Minimum and Maximum). Find numbers which are divisible by 5 and multiple of 9 between the given range of minimum and maximum numbers.
b. Write a Python program to count the number of prime, even and odd numbers from a series of numbers.

Problem Statement 5: (Lists)

a. Write a Python program to find a list of Strings with exactly two occurrences of “MIT” and at least one occurrence of “ADT”.
   Input: [“MIT”, “SOE”, “MIT” , “ADTU”, “ADT”, “Loni”, “Design”, “Food”, “Technology”]
b. Extend the program for the following integer inputs.
   Input: [100, 35, 23, 100, 45, 89, 90] //Exactly 2 occurrences of 100 and at least 2 occurrences of 90
   Input: [90, 110, 130, 150, 170, 200, 200] //At Least 2 occurrences of 200 and exactly one occurrence of 130
c. Write a Python program that accepts a list of integers and checks the length and the third element. Return true if the length of the list is more than 10 and the third element occurs twice in the said list.
d. Sort the list in ascending order.

Problem Statement 6: (Dictionaries)

e2 Write a Python program to convert two lists into a dictionary in a way that item from list1 is the key and item from list2 is the value$

b. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, square of x)

c. Write a Python program to create a dictionary storing movie details such as - (Movie name: show timings) and perform the following operations

   i. Add a new movie
   ii. Display movies available at 9 pm
   iii. Remove the details of the specific movie
   iv. Removes the last movie details

—---

Unit 3: Functions and Modules

Introduction, Defining user defined functions, Parameters, Function documentation, Keyword and Optional Parameters, Passing Collections to a Function, Variable Number of Arguments, parameter scope, Passing Functions to a Function, Mapping Functions in a Dictionary, Lambda, Functional programming tools - filter(), map(), and reduce()

Modules - Standard Modules such as sys, math, time, random, etc., dir Function

—---

Problem Statement 7: (Functions)

Write a Python function to -

- find the Max of three numbers
- multiply all the numbers in a list
- create a list of all even numbers between 19 and 88
- display the current date and time and get the Python version

Problem Statement 8: (Functions)

Write a program to build a simple Employee and Project management System using Python functions which can perform the following operations:

1. Accept [Project name, Team members, Technologies used, Project completion deadline]
2. Display - displays the details of every Project
3. Search - Look for a specific employee and the project on which he or she is working
4. Delete - delete a particular record (based on either project/employee
5. Update - Update the team members for a particular project

—---

Unit 4: Classes, Objects, and Regular Expressions

Classes in Python, Principles of Object Orientation, Creating Classes, Instance Methods, File Organization, Special Methods, Class Variables, Inheritance, Polymorphism, Type Identification, Custom Exception Classes

Regular Expressions - Concept of regular expression, various types of regular expressions

Packages: Importing package, creating package, examples

—---

Problem Statement 9: (Classes)

Write a program to build the animal shelter (holding cats, dogs, and rabbit) operations using Python class. People can adopt the animals based on their admission time. Create a data structure to perform various operations such as - new entry in the shelter, adopt a dog, adopt a cat, adopt a rabbit, adopt any animal, display the shelter information

Problem Statement 10: (Classes)

Write a program to ATM machine operation utilizing classe, inheritance concepts for following operations (Assume customer has savings bank account with respective bank:

a. Check Balance
b. Withdraw Amount
c. Deposit Amount
d. Mini Statement
e. Change Pin

—---

Unit 5: Input/Output and Exception Handling

Introduction, Data Streams, Access Modes, Writing Data to a File, Reading Data From a File, Additional File Methods, Using Pipes as Data Streams

Working with Directories - Removing files and directories, copying and renaming files, splitting pathnames, creating and moving directories

Handling IO Exceptions, Metadata, Errors, Run Time Errors, The Exception Model - Exception Hierarchy, Handling Multiple Exceptions, User-defined Exceptions.

—---

Problem Statement 11: (Files)

Write a Python GUI program for the student registration and store the details in the file; Allow the users to read the details from the file.

Problem Statement 12:(Exception handling)

Write a generic program to handle exception generated in following scenarios:

- Division by Zero
- Accessing a file which does not exist.
- Addition of two incompatible types
- Trying to access a nonexistent index of a sequence

—---

Course Outcomes:

Students will be able -

- To apply the principles of python programming.
- To create applications using python programming.
- To determine the methods to create and develop Python programs by utilizing the data structures like lists, dictionaries, tuples and sets
- To handle run-time and application errors

Reference Books:

1. Dive into Python, Mike
2. Learning Python, 4th Edition by Mark Lutz
3. Programming Python, 4th Edition by Mark Lutz
4. Object-oriented Programming in Python, Michael H. Goldwasser, David Letscher, Pearson Prentice Hall-2008