

COURSE NAME: Python Programming and Minor Project Lab

Total T-L Hours= 60

T-L hours/week = 4

Total Marks = 75

Internal Marks = 45

External Marks = 30

Course Outcome 1	Interpret basic constructs of python programming	Teach Hrs	Marks
Learning Outcome 1	Describe steps to setup python development environment, features and modules.	5	05
Contents	Features and applications of python, steps of installing/setting python, Integrated Development and Learning Environment (IDLE), Basic structure of python program, creating, editing, executing python program with IDLE, concept of modules in python, from/import statement, concept of PYTHONPATH and sys.path variables, Renaming/alias a module, python package-management system (pip), Namespaces and Scope in Python, built-in/global/local namespace, datetime module.		
Method of Assessment	Lab Assessment (File)		
Learning Outcome 2	Explain different data types, operators, comments and user defined functions in Python.	6	10
Contents	Keywords, variables and comments. Concept of input and output. Various data types, type conversion. Unary, binary and ternary operators. Arithmetic operators, Assignment operators, Relational operators, Logical operators, Bitwise operators. User defined functions.		
Method of Assessment	Quiz		
Learning Outcome 3	Write programs using basic constructs.	6	10
Contents	Working on conditional statements (if-else-elseif) and Iterative statements (for, while) and flow-charts. Loop control statements - break, continue, pass. Use math module.		
Method of Assessment	Lab Assessment (File)		
Course Outcome 2	Develop programs using structure types, exception handling and file handling.	Teach Hrs	Marks
Learning Outcome 4	Use built in functions to manipulate strings and lists.	6	10
Contents	Create list, indexing in list, access list items, add, remove and modify item value. Iterating over list, check if item exists in list, length of list, copy list. Join two lists, sort, and reverse a list. Working with strings, multiline strings. String indexing, string slicing, string length, escape character, search and split operation in string, convert object to string using str ().		
Method of Assessment	Programming (External)		
Learning Outcome 5	Write programs using tuples, dictionaries and sets.	6	10

Contents	<p>Create tuple, indexing in tuple, access items of tuple. Converting tuple to list. Iterating over tuple, check if an item exists in tuple. Tuple length, join two tuples.</p> <p>Create set, access items of set, add items to set, length of set. Remove an item in set, join two sets, and update set. Union operation of set.</p> <p>Create dictionary, keys-value pair in dictionary. Add, remove and access items of the dictionary. Change item value. Iteration over dictionary, check if key exists, length of dictionary, copy dictionary.</p>		
Method of Assessment	Lab Assessment (File)		
Learning Outcome 6	Use exception handling, built in libraries to perform file/directory related operations and numpy module to manipulate arrays.	7	05
Contents	<p>Need for Exception Handling. Try, except statement. Raise exception, user defined exceptions.</p> <p>Perform basic file/directories related operations such as - create, copy, move, or rename, reading and writing file contents.</p> <p>numpy module - create arrays, indexing in array, accessing arrays, slicing arrays, modify and copy array. Create view of array. Iterating over array, join two arrays, split arrays, searching, sorting and sort method.</p>		
Method of Assessment	Quiz		
Course Outcome 3	Apply the concepts of object-oriented programming and GUI designing.	Teach Hrs	Marks
Learning Outcome 7	Write programs using classes, objects, constructors and inheritance.	6	05
Contents	<p>Basic syntax and structure of a Class. Declare/define Variables and methods in a class. Class or Static Variables in class. Creating objects and access class members using dot (.) operator. Constructors in class, default constructor, parameterized constructor, self-keyword, destructors in python and inheritance.</p>		
Method of Assessment	Programming (External)		
Learning Outcome 8	Develop GUI using tkinter interface.	6	05
Contents	<p>Basics of tkinter module. Creating main window, configure properties of main window such as -title, size. Tk(), mainloop() methods. Adding basic widgets - Button, Label, Entry, Text, Canvas, Frames. Drawing shapes such as - lines, oval, rectangle etc on canvas.</p>		
Method of Assessment	Lab Assessment (File)		
Learning Outcome 9	Develop problem solving ability and use this knowledge of programming to solve industry projects.	12	15
Contents	<p>Use functions, classes, basic constructs, standard built-in packages and other Python libraries to develop real-world user-friendly applications such as games (snake game using turtle package, tic-tac-toe etc.), creating English-dictionary, calculator, alarm-clock, student-management system projects, etc.</p>		
Method of Assessment	External		

The project report should include the following:

- Cover page: (use required format if provided by faculty)
- Certificates
- Acknowledgement
- Abstract
- Contents
- Introduction
- Literature review
- Problem formulation
- Project methodology
- Flowcharts/DFDs/ERDs
- Algorithms/Project Code
- Outputs/Screenshots
- Conclusions and Future scope
- References
- Appendix
- Plagiarism report

Suggested List of Experiments:

S.No.	Experiment
1.	Setting up environment/ IDE to create, edit and run Python Programs
2.	Write a program that will output “HELLO WORLD” on the screen.
3.	Write a program to read input from keyboard of various data types and print them.
4.	Use arithmetic, logical, and bitwise operators in your programs.
5.	Write a program to demonstrate, type casting, global variables, comments and user defined functions, errors due to indentation.
6.	Write programs to illustrate the use of conditional and looping statements
7.	Write programs to illustrate various operations on list, tuples, dictionaries and sets.
8.	Implement programs to perform various operations on string using string libraries.
9.	Implement programs to demonstrate various math built-in functions.
10.	Develop user defined python functions to demonstrate function that takes arguments and return values.
11.	Write programs using classes, objects, constructors, and inheritance.
12.	Write programs to demonstrate use of built-in packages (ex. Numpy, pandas, matplotlib etc.)
13.	Implement programs to demonstrate file related operations (ex. Create, modify, delete, read, write etc.)
14.	Implement programs to raise exception and perform exception handling
15.	Write a program to create a basic window GUI using tkinter interface.