Course Title: **Python Programming**

Course No. : ICT. Ed. 477 | Nature of course: Theoretical + Practical

Level: Bachelor | Credit hours: 3 (2T+1P)

Semester: Seventh | Teaching hours: 64 (32T+32P)

---

1. Course Description

The aim of the course is to impart knowledge of the basic concepts of python programming and to help the students build skills for solving problems using it. It provides the students with the basic features of the language such as data types, operators, control structure, list, dictionaries, sets, tuples, string manipulation, functions, exception and file handling which are the common features of programming languages. It also provides knowledge about object oriented paradigm, database programming and building graphical user interfaces. Students are more engaged in laboratory work to exaction of programing experiments rather than theoretical concept.

2. General Objectives of the Course

Following are the general objective of this course:

- To make the student knowledgeable about the python programming concept.
- To enable the student in implement the essential programming concepts and methods in practices.
- To acquaint the student in organization of data in lists, dictionaries, sets and tuples.
- To explore the database programming, graphical user interface programming using python.
- To provide the students with the skills of object orientated programming to solve the real world problems.

3. Specific Objectives and Contents

Specific Objectives	Contents
• Write and Execute Python Program • Describe basic structure of python program, data types, variables, operators, comments and constants. • Perform input and output operation	**Unit 1: Python Programming Fundamentals [8]** 1.1 Python Introduction 1.2 Data Types and Type Conversion 1.3 Comments 1.4 Variables, Constants, Operators and Performing Calculations 1.5 Reading Input from Keyboard 1.6 Print function, Displaying Formatted Output with F-strings

• Explore print function and display formatted output with f-string	**Practical Works** • *Write program to illustrate variables, constants, data types and type conversion.* • *Write program to demonstrate different types of operators available in python and perform calculations.* • *Write program to make use of I/O functions.*
• Explain control statements. • Write decision making problems using if, match, break and continue statements • Apply different types of loop and make distinction among them through program. • Discuss different problems and how they are transformed to programs using nested loop and infinite loop.	**Unit 2: Control Statements in Python [8]** 2.1 if statement 2.2 match statement 2.3 break statement 2.4 continue statement 2.5 Loop statement 2.5.1 while 2.5.2 for 2.6 Nested loop 2.7 Infinite loop **Practical Works** • *Write program to apply if, match, break and continue statements for decision making.* • *Write program to utilize different loop statements to solve meaningful problems.* • *Write program to demonstrate input validation using loop.* • *Write program to create different patterns using nested loop.* • *Write program to make use of infinite loop.*
• Describe List, Tuples, Dictionary, Sets and Strings in python • Elaborate List Comprehension, Dictionary operations, Set	**Unit 3: List, Tuple, Dictionaries, Sets and Strings [16]** 3.1 Introduction to Lists - List Slicing, in operator, List Methods : append, index, insert, sort, remove, reverse, min, max 3.2 List Comprehension 3.3 Two-Dimensional Lists 3.4 Tuples

Operations, Tuples Operations, List slicing. • Solve simple computing problems using List methods. • Use different string methods to manipulate strings. • Describe Searching and Sorting problem.	3.5 Dictionaries - Creating dictionary, retrieving, adding and removing elements 3.6 Sets - Creating Set, Adding and Removing Elements - Set Operations: union, intersection, difference 3.7 Strings - String Operations : Slicing, Testing, Searching, and Manipulating **<u>Practical Works</u>** • *Write program to create list, add elements in list, remove elements from list and display list items.* • *Write program to make use of list slicing concept to display elements of list.* • *Write program to elaborate different list methods.* • *Write program to apply list comprehension.* • *Write program to illustrate two-dimensional list.* • *Write program to create tuple, add elements in tuple, remove elements from tuple and display tuple items.* • *Write program to create dictionary, add elements in dictionary, remove elements from dictionary and display dictionary items.* • *Write program to create set, add elements in set, remove elements from set and display set items.* • *Write program to perform set operations.* • *Write program to make use of string manipulation methods and also perform different string operations.*
• Define object oriented paradigm with python • Demonstrate class and object with data hiding concept • Describe the use of self-keyword in programs.	**Unit 4: Object Oriented Programming with Python [10]** 4.1 Class and Object 4.2 __init_ method 4.3 self keyword 4.4 Inheritance 4.5 Polymorphism and Data Hiding

• Compare and contrast different types of inheritance. • Elaborate polymorphism concept	**Practical Works** • *Write program to elaborate object oriented concept with simple examples.* • *Write program to make use of __init_ method to initialize objects.* • *Write program to apply different types of inheritance.* • *Write program to elaborate polymorphism and data hiding concept.*
• Clarify the concept of functions. • Create function with arguments, without arguments, returning values. • Describe exception and how to handle them in programs. • Explain the use of file in program. • Demonstrate the file operations with examples.	**Unit 5: Function, Exception and File Handling [10]** 5.1 Introduction to Functions 5.2 Defining and Calling Function 5.3 Passing Arguments to Functions 5.4 Value-Returning Functions 5.5 Introduction to File Input and Output 5.6 Using Loops to Process Files 5.7 Exception Handling **Practical Works** • *Write program to divide work in functions.* • *Write different variety of functions: function with arguments, value returning function, function without arguments.* • *Write program to store output in file.* • *Write program to read input from file.* • *Write program to handle different types of exception.*
• Discuss the use of database and GUI programming. • Demonstrate CRUD operation in database. • Design simple GUI with frames and widgets. • Perform simple calculations using GUI.	**Unit 6: Database and GUI Programming [12]** 6.1 Opening and Closing Database Connection with SQLite 6.2 Creating and Deleting Tables 6.3 Adding Data to a Table 6.4 CRUD Operations 6.5 Using the tkinter Module 6.6 Working with Widgets

• Draw different geometrical shapes using canvas	- Displaying Text with Label, Button, Info Dialog Boxes, Getting Input with the Entry, Using Labels as Output Fields, Radio and Check Buttons 6.7 Organizing Widgets with Frames 6.8 Drawing Shapes with Canvas Widget **<u>Practical Works</u>** • *Write program to establish connection with database and create or delete database and table.* • *Write program to store data in database and manipulate the data.* • *Write program to perform CRUD operation in database.* • *Write program to create simple GUI with widgets: label, text entry, radio buttons, check buttons* • *Write program to organize the different widgets with frame to create attractive designs.* • *Write program to draw different geometrical shapes using canvas widget.*

Note: The figures in square brackets indicate approximate teaching hours allotted to respective units.

4. General Instructional Techniques

Lecture preferably with the use of multi-media projector, demonstration, practical classes, discussion, and brain storming in all units as far as practicable.

4.1 Specific Instructional Techniques

Demonstration is an essential instructional technique for all units in this course during teaching-learning process. Specifically, demonstration with practical works will be specific instructional technique in this course. The details of suggested instructional techniques are presented below:

Units	Activities
Unit 1 to 6	• Code writing activity is performed to elaborate each units concepts • Monitoring of students' work by reaching each student and providing feedback for improvement • Presentation by students followed by peers' comments and teacher's feedback • Demonstration by the teacher on practical works mentioned in each unit

	• Lab work individually or in pairs is assigned by the teacher to understand each unit • Assignment should be assigned to prepare lab report/project report for individual student

5. Evaluation

Evaluation of students' performance is divided into parts: Internal assessment and internal and external practical examination and theoretical examinations. The distribution of points is given below:

Internal Assessment	Internal and External Practical Exam/Viva	Semester Examination (Theoretical exam)	Total Points
40 Points	20 Points	40 Points	100 Points

***Note**: Students must pass separately in internal assessment, external practical exam and semester examination.*

5.1 Internal Assessment (20 Points)

Internal assessment will be conducted by subject teacher based on following criteria:

Attendance and learning Activities	5 points
First assignment (Written assignment)	5 points
Second assignment (Project work with presentation)	10 points
Total	20 points

5.2 Semester Examination (40 Points)

Examination Division, Dean office will conduct final examination at the end of semester.

Objective question (Multiple choice questions 10 x 1 point)	10 Points
Short answer questions (6 questions x 5 marks)	30 Points
Total points	40

5.3 Practical Exam/Viva (20 Points)

Internal assessment (Record Book-4 points, Project work Presentation- 2, Internal Practical Test-2 Points)	Semester final examination	Total
8 Points	12 Points	20 Points

6. Recommended Books and References materials (including relevant published articles in national and international journals)

6.1 Prescribed Textbook

Tony Gaddis, T. (2021). *Starting out with Python (5th Ed.).* Pearson

6.2 Recommended Books

Hetland, M.L. & Nelli, F. (2024). *Beginning Python from Novice to Professional (4th Ed.),* Apress

Murugesh, T.S., Vasudevan, S.K. & Pulari, S.R. (2024). *Python: A Practical Learning Approach (1st Ed.),* CRC Press

Zuckarelli, J.L. (2024). *Learn coding with Python and JavaScript A practical introduction for beginners (1st Ed.),* Springer

Barry, P. (2023). *Head First Python: A Learner's Guide to the Fundamentals of Python Programming (3rd Ed.),* O'Reilly Media

Liu, M. (2021). *Make Python Talk Build Apps with Voice Control and Speech Recognition (1st Ed.),* No Starch Press