# GUJARAT TECHNOLOGICAL UNIVERSITY

**Program Name: Diploma Engineering**

**Level: Diploma**

**Branch: Electronics and Communication Engineering**

**Subject Code : DI04011041**

**Subject Name : Python Programming**

| | |
|---|---|
| **w. e. f. Academic Year:** | 2025-26 |
| **Semester:** | 4th |
| **Category of the Course:** | Professional Elective - II |

| | |
|---|---|
| **Prerequisite:** | Basic computer skills |
| **Rationale:** | Programming skills using Python Language have become prevalent and increasingly popular in the Information Technology domain. The Python programming language is one of the most popular programming languages worldwide. The Python Programming for Electronics & Communication (EC) course emphasizes the use of this popular programming platform in multiple domains. Python is a modern language for writing compact codes specifically for programming Server-side web apps, Data Analytics and Machine Learning, an important Artificial Intelligence domain. Furthermore, Python has gained popularity in scientific computing, production tools and game programming. The following course focuses on developing Python Programs for programming tasks where the students are encouraged to create basic applications using different open source tools. At the end of the course, the student will be developing adequate basic programming skills using python language. |

**Course Outcome:**

After Completion of the Course, Student will able to:

| No | Course Outcomes | RBT Level |
|---|---|---|
| 01 | Prepare flowchart and algorithm for solving computing problems. | R,U,A |
| 02 | Develop python programs to solve simple problems. | R,U,A |
| 03 | Apply control structure feature of python for developing programs. | R,U,A |
| 04 | Develop Python Programs using built-in functions, modules and libraries | R,U,A |
| 05 | Develop python programs using dictionary, list, string, set and tuple manipulation functions. | R,U,A |

**Revised Bloom's Taxonomy (RBT)*

**Teaching and Examination Scheme:**

| Teaching Scheme (in Hours) | | | Total Credits L+T+ (PR/2) | Assessment Pattern and Marks | | | | Total Marks |
|---|---|---|---|---|---|---|---|---|
| | | | | Theory | | Tutorial / Practical | | |
| L | T | PR | C | ESE(E) | PA(M) | PA(I) | ESE(V) | |
| 3 | 0 | 2 | 4 | 70 | 30 | 20 | 30 | 150 |

**Course Content:**

| Unit No. | Content | No. of Hours | % of Weightage |
|---|---|---|---|
| **1. Problem Solving using Flowchart and Algorithm** | 1.1 Introduction, Steps for problem- solving, Algorithm and its characteristics, Importance of flowchart and algorithm<br>1.2 Symbolic representation of a flowchart, Limitations of flowchart Flow of control<br>1.3 Problem solving using pseudocode | 05 | 10 |
| **2. Introduction to Python** | 2.1 Introduction to python, Python features, Applications of python programming<br>2.2 Python installation<br>2.3 Basic structure of python program, Keywords, identifiers, and variables, Data types, Operators<br>2.4 Type Conversion | 06 | 14 |
| **3. Flow of Control** | 3.1 Introduction to Flow of Control<br>3.2 Selection<br>If statement<br>Elif statement<br>Nested if statement<br>3.3 Repetition<br>For loop<br>While loop<br>Nested loop<br>3.4 Break and Continue Statements | 12 | 23 |
| **4.** | 4.1 Introduction to Functions | 08 | 20 |

| | | | |
|---|---|---|---|
| **Functions** | User Defined Functions<br>Arguments and Parameters<br>4.2 Scope of a Variable<br>Global Variable<br>Local Variable<br>4.3 Python Standard Library<br>Built-in functions<br>Input or output - input(), print()<br>Mathematical Functions- abs(), divmod(), max(), min(), pow(), sum()<br>Module - Math module, Random module, Statistics module | | |
| **5. Dictionary, List, Set, String and Tuple** | 5.1 Introduction to String, String Operations, Traversing a String<br>5.2 String Methods and Built-in Functions<br>5.3 Introduction to List and its Operations<br>5.4 List Methods and Built-in Functions<br>5.5 Set Built-in Functions, Create a Set, Accessing Python Sets, Delete from set, Update set<br>5.6 Python Set Operations<br>5.7 Tuple Built-in Functions, Creating Tuples, Accessing Tuple, Iterate over tuple and Slicing tuple<br>5.8 Python Tuple Operations, Functions and Methods<br>5.9 Dictionary Built-in Functions, Creating Dictionary, Accessing Items, Add, Update, Remove in Dictionary<br>5.10Built-In Dictionary Methods and functions | 14 | 33 |
| | **Total** | **45** | **100** |

**Distribution of Theory Marks (in %):**

| Distribution of Theory Marks (in %) | | | | | |
|---|---|---|---|---|---|
| **R Level** | **U Level** | **A Level** | **N Level** | **E Level** | **C Level** |
| **22** | **28** | **50** | -- | -- | -- |

*Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)*

**References/Suggested Learning Resources:**

**(a) Books:**

1. Learn Programming in Python with Cody Jackson by Cody Jackson, Packt Publishing, 2018, ISBN : 9781789531947
2. Python Basics: A Practical Introduction to Python 3 by David Amos, Dan Bader et. al. Real Python, 2021 ISBN : 9781775093329
3. Introduction to Problem Solving with Python by E. Balagurusamy Mc Graw Hill India, New Delhi, 2017 ISBN: 9789352602582
4. Beginning Python by James Payne, Wiley, 2010 ISBN: 9780470414637
5. Think Python by Allen Downey, O'Reilly, USA, 2016, ISBN : 978- 9352134755

**(b) Open source software and website:**

1. www.python.org
2. https://www.w3schools.com/python/python_intro.asp
3. https://www.geeksforgeeks.org/python-programming-language-tutorial/
4. www.learnpython.org
5. www.hackr.io/tutorials/learn-python
6. www.sololearn.com/learning/1073
7. www.nptel.iitm.ac.in
8. https://scratch.mit.edu/

**Suggested Course Practical List:**

| S. No. | Practical Outcomes (PrOs) | Unit No. | Approx .Hrs. required |
|---|---|---|---|
| 1 | Prepare flowchart and algorithm for a given problem.(Following arethe sample programs. Faculty can select any other similar programsfor the practice of the students.)<br>i. Find the sum of two given numbers.<br>ii. Find a maximum out of two given numbers.<br>iii. Find whether a given number is odd or even.<br>iv. Find a maximum out of three given numbers. | I | 02 |
| 2 | a) Write a Program to print your name, mobile number, and date of birth.<br>b) Develop a Program to identify data-types in python. | II | 02 |
| 3 | a) Create a Program to read three numbers from the user and find the average of the numbers.<br>b) Develop a Program that can calculate simple interest and compound interest on given data.<br>c) Write a Program to convert temperature from Fahrenheit to Celsius unit using eq: C=(F-32)/1.8 | II | 02 |

| | | | |
|---|---|---|---|
| 4 | a) Identify whether the scanned number is even or odd and print an appropriate message.<br>b) Create a Program to find a maximum number among the given three numbers. | II I | 02 |
| 5 | Develop a Program to demonstrate the basic functionalities of a standard calculator. | II I | 02 |
| 6 | a) Write a Python program to print 1 to 10 numbers using loops.<br>b) Write a Program to show whether the entered number is prime or not. | II I | 02 |
| 7 | Write a Program to display the following patterns.<br>A) B)<br>1 * * * * *<br>1 2 * * * *<br>1 2 3 * * *<br>1 2 3 4 * *<br>1 2 3 4 5 * | II I | 02 |
| 8 | Create a User-defined function to print the Fibonacci series of 0 to N numbers. (Where N is an integer number and passed as an argument) | I V | 02 |
| 9 | a) Develop a User-defined function to find the factorial of a given number.<br>b) Write a Program using the function that reverses the entered value. | I V | 02 |
| 10 | Write a Program that determines whether a given number is an 'Armstrong number' or a palindrome using a user-defined function. | I V | 02 |
| 11 | a) Write a Program to find the length of a string.<br>b) Write a Program to reverse words in a given sentence.<br>c) Write a Program to check if a substring is present in a given string.<br>d) Write a Program to count and display the number of vowels, consonants, uppercase, lowercase characters in a string. | V | 02 |
| 12 | Develop Programs to perform the following list operations.<br>a) To swap given two elements in a list.<br>b) To find the sum of elements in a list.<br>c) To check if an element exists in a given list.<br>d) To find the smallest and largest element in a given list. | V | 02 |

| | | | |
|---|---|---|---|
| 13 | a) Given a List saved in variable: a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. Write one line of Python that takes this list and makes a new list that has only the even elements of this list in it.<br>b) Create a List containing the square of all odd numbers from range 1 to 10.<br>c) Create a List of prime and non-prime numbers in range 1 to 50. | V | 02 |
| 14 | a) Write a Program to demonstrate some useful set functions and operations.<br>b) Write a Program to demonstrate some useful Tuple functions and operations. | V | 02 |
| 15 | a) Write a Program to demonstrate the dictionaries functions and operations.<br>b) Create a Dictionary with the roll number, name, and marks of n students in a class and display the names of students who have scored marks above 75. | V | 02 |
| | **Total** | | **30** |

**List of Laboratory/Learning Resources Required:**

| S. No. | Equipment Name with Broad Specifications | PrO. No. |
|---|---|---|
| 1 | Computer system with operating system: Windows 7 or higher Ver., macOS, and Linux, with 4GB or higher RAM, Python versions:2.7.X, 3.6.X | All |
| 2 | Python IDEs and Code Editors<br>Open Source : IDLE, Jupyter | 2 to 8 |

**Suggested Project List:**

***Only one micro-project*** is planned to be undertaken by a student that needs to be assigned to him/her in the beginning of the semester. In the first four semesters, the micro-project are group-based. However, in the fifth and sixth semesters, it should be preferably be ***individually*** undertaken to build up the skill and confidence in every student to become problem solver so that s/he contributes to the projects of the industry. In special situations where groups have to be formed for micro-projects, the number of students in the group should ***not exceed three.*** The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. Each student will have to maintain a dated work diary consisting of individual contributions in the project work and give a seminar presentation of it before submission. The total duration of the micro-

project should not be less than ***16 (sixteen) student engagement hours*** during the course. The student ought to submit a micro-project by the end of the semester to develop the industry oriented COs.

A suggestive list of micro-projects is given here. This has to match the competency and the COs. Similar micro-projects could be added by the concerned course teacher:

- **Case Study 1:** Compare three various tools functionality for python programming in the EC domain
- **Case Study 2:** List out features of latest python version and compare with older two different versions.
- **Case Study 3:** Study math and random module of python and explain real life usage with examples. List their importance in Information Communication Technology
- **Case Study :** Practice logic building using scratch tool available free and online by mit

**********