GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)

Competency-focused Outcome-based Green Curriculum-2021 (COGC-2021)
Semester-V

Course Title: OOPS & Python Programming
(Course Code: 4351108)

Diploma programmer in which this course is offered	Semester in which offered
Electronics & Communication Engineering	5th Semester

1. RATIONALE
A course on Object-Oriented Programming (OOPS) and Python Programming can be beneficial for students as it covers basic and advanced programming concepts and provides a foundation for software development. The course cover a variety of topics, including basic programming concepts, advanced OOPS concepts. Additionally, the course can provide best practices and tips for writing Python code, debugging, and testing code, and avoiding common mistakes. Overall, the course can equip students with the skills and knowledge needed to succeed in a variety of programming roles.

2. COMPETENCY
The purpose of this course is to help the student to attain the following industry identified competency through various teaching learning experiences:

- **Develop program using Object-Oriented Python Programming (OOPS) to solve given problem.**

3. COURSE OUTCOMES (COs)
The practical exercises, the underpinning knowledge and the relevant soft skills associated with the identified competency are to be developed in the student for the achievement of the following COs:

- Demonstrate proficiency in Python programming fundamentals, including data types, control structures, loops, input-output functions.
- Develop python programs by applying data structures – list, dictionary, tuple, set and strings concepts.
- Design Python functions and modules.
- Apply object-oriented programming principles using Python.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme (In Hours)			Total Credits (L+T+P/2)	Examination Scheme				
				Theory Marks		Practical Marks		Total Marks
L	T	P	C	CA	ESE	CA	ESE	
2	0	2	3	30*	70	25	25	150

(*): Out of 30 marks under the theory CA, 10 marks are for assessment of the micro-project to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessing the attainment of the cognitive domain UOs required for the attainment of the COs.
***Legends:** **L**-Lecture; **T** – Tutorial/Teacher Guided Theory Practice; **P** - Practical; **C** – Credit, **CA** - Continuous Assessment; **ESE** - End Semester Examination.*

5. SUGGESTED PRACTICAL EXERCISES

The following practical outcomes (PrOs) are the subcomponents of the Course Outcomes (Cos). Some of the **PrOs** marked **'*'** are compulsory, as they are crucial for that particular CO at the 'Precision Level' of Dave's Taxonomy related to 'Psychomotor Domain'.

Sr. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
1	Install & configure python software.	I	2
2	(A) Create a program to print your name, mobile number, and date of birth. (B) Create a program to print following sentence. *John said, "There's an elephant outside the window."*	I	2
3	(A) Create a program to read three numbers from the user and find the average of the numbers. (B) Create a program to convert temperature from Fahrenheit to Celsius unit using eq: C=(F-32)/1.8 (C) Create a program that can calculate simple interest and compound interest on given data.	I	2
4	(A) Create a program to identify whether the scanned number is even or odd and print an appropriate message. (B) Create a program to find a maximum number among the given three numbers. (C) A year is a leap year if it is divisible by 4, except that years divisible by 100 are not leap years unless they are also divisible by 400. Create a program that asks the user for a year and prints out whether it is a leap year or not. (D) Suppose that scores 90 and above are A grade, scores in the 80s are B grade, 70s are C grade , 60s are D grade , and anything below 60 is an F grade. Create a program to find grade according to scores entered by user.		2
5	(A) Create a python program to print 1 to 10 numbers using loops. (B) Create a program to find the sum of all the positive numbers entered by the user. As soon as the user enters a negative number, stop taking in any further input from the user and display the sum. (C) Create a program to display the following patterns. 1 | * * * * * 2 2 | * * * * 3 3 3 | * * * 4 4 4 4 | * * 5 5 5 5 5 | * (D) Create a program to print prime numbers between 1 to N.		2

Sr. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
6	(A) Create a program to find sum of all elements in a list using for loop. (B) Create a program to increment each element of list by one. (C) Create a program to check if an element exists in a given list. (D) 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.		2
7	(A) Write a program to input names of n employees and store them in a tuple. Also, input a name from the user and find if this employee is present in the tuple or not. (B) Write a program to read email IDs of n number of students and store them in a tuple. Create two new tuples, one to store only the usernames from the email IDs and second to store domain names from the email ids. Print all three tuples at the end of the program.		2
8	Write a Program to create three sets A={1,5,6,3,7}, B={1,8},C={9,3,5,6} and perform following operations. (1) Add two elements 3 and 4 to set B. (2) Remove 9 from set C (3) Find union of A and C (4) Find intersection of A and B (5) Find difference and symmetric difference of A and C (6) Check if set A and B are disjoint or not (7) Check if A is subset of C or not (8) Check if B is superset of A or not		2
9	(A) 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. (B) Write a program to count the number of times a character appears in a given string using a dictionary. (C) Write a user-defined function to convert a number entered by the user into its corresponding number in words. For example, if the input is 789 then the output should be 'Seven Eight Nine'		2
10	(A) Write a program to find the length of a string excluding spaces. (B) Write a program that asks the user for a string and prints out the location of each 'a' in the string. (C) Write a program to input a string from the user and print it in the reverse order without creating a new string.		2

Sr. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
11	(A) 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) (B) Create user-defined function to check if given number is prime or not. (C) Create user defined function to check if given string is palindrome or not. (D) Create a program to define a module to find the area and circumference of a circle. i. import the module to another program. ii. import a specific function from a module to another program. (E) Write a program to plot sine and cosine wave in one plot using *numpy* and *matplotlib* module.		4
12	(A) Design a class Complex for adding the two complex numbers and also show the use of constructor. (B) Design a class for single level inheritance using public and private type derivation. (C) Write a Python program to demonstrate method overriding using inheritance. (D) Implement multiple and hierarchical inheritance.		4
			28 Hrs.

Note

i. *More **Practical Exercises** can be designed and offered by the respective course teacher to develop the industry relevant skills/outcomes to match the COs. The above table is only a suggestive list.*

ii. *The following are some **sample** 'Process' and 'Product' related skills (more may be added/deleted depending on the course) that occur in the above listed **Practical Exercises** of this course required which are embedded in the COs and ultimately the competency.*

Sr. No.	Sample Performance Indicators for the PrOs	Weightage in %
1	Identify suitable approach to implement logic	25
2	Use pre-built packages/functions	20
3	Use python concepts to implement efficient program	25
4	Follow different input test cases to check output	10
5	Identify and mend coding errors in a program / Interpret the result and conclude	20
Total		**100**

5. MAJOR EQUIPMENT/ INSTRUMENTS AND SOFTWARE REQUIRED

This major equipment with broad specifications for the PrOs is a guide to procure them by the administrators to use in uniformity of practical's in all institutions across the state.

Sr.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: 3.7.X	All
2	Python IDEs and Code Editors Open Source : IDLE/PyCharm/Spyder/Jupyter	

6. AFFECTIVE DOMAIN OUTCOMES

The following ***sample*** Affective Domain Outcomes (ADOs) are embedded in many of the above-mentioned COs and PrOs. More could be added to fulfill the development of this course competency.

a. Work as a leader/a team member (while doing a micro-project).
b. Follow safety practices.
c. Maintain tools and equipments.
d. Adhere to ethical practices.

The ADOs are best developed through the laboratory/field-based exercises. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

i. 'Valuing Level' in 1st year
ii. 'Organization Level' in 2nd year.
iii. 'Characterization Level' in 3rd year.

8. UNDERPINNING THEORY

The major underpinning theory is given below based on the higher level UOs of *Revised Bloom's taxonomy* that are formulated for development of the COs and competency. If required, more such UOs could be included by the course teacher to focus on attainment of COs and competency.

Unit	Unit Outcomes (UOs) (4 to 6 UOs at different levels)	Topics and Sub-topics
Unit-I Basics of Python	1.a Install and configure Python. 1.b Describe the different data types available in Python and use them. 1.c Convert between different data types using type casting in Python. 1.d Describe the need for control structures in programming. 1.e Use decision making structures, different types of loop, break, continue	**1.1** Introduction to Python, Python Features, Python Applications **1.2** Installing Python **1.3** Basic Structure of Python Program, Keywords, Identifiers, Data types, Variables, Operators, Type Casting **1.4** Input-Output functions: input, print **1.5** Introduction to Control Structures **1.6** Decision Making Structures: if, if-else statements, Nested if-else and if-elif-else statements **1.7** Loops : for loop, while loop, Nested loops,

Unit	Unit Outcomes (UOs) (4 to 6 UOs at different levels)	Topics and Sub-topics
	and pass statements.	**1.8** break, continue and pass statements
Unit-II Lists, Tuples, Sets, Dictionaries and String	2.a Develop programs using lists, tuples, sets, dictionaries, and string. 2.b Use various operations on lists, tuples, sets, dictionaries, and string.	**2.1** Lists and operations on Lists **2.2** Tuples and operations on Tuples **2.3** Sets and operations on Sets **2.4** Dictionaries and operations on Dictionaries **2.5** String and operations on strings
Unit-III Functions and Modules	3.a Create user-defined functions in Python. 3.b Explain the concepts of global and local variables in Python. 3.c Use math, random and statistics module in programs. 3.d Create and import user-defined modules in Python.	**3.1** Introduction to Functions -User defined functions -Arguments and Parameters **3.2** Scope of Variable: Global and Local Variable **3.3** Module: math, random, statistics **3.4** Creating user defined module
Unit-IV Object Oriented Programming	4.a Define Object-Oriented Programming and its importance in Python. 4.b Create objects from a class in Python. 4.c Explain the difference between these types of methods and when to use each of them. 4.d Use encapsulation to hide the implementation details of a class from the user. 4.e Implement single, multiple, multi-level, hierarchical, and hybrid inheritance in Python programs. 4.f Use polymorphism to write flexible and reusable code in Python.	**4.1** Oops Concepts **4.2** Class and Objects **4.3** Constructors **4.4** Types of methods:Instance method, Class method, static method **4.5** Data Encapsulation **4.6** Inheritance- single, multiple, multi level, hierarchical, hybrid **4.7** Polymorphism through inheritance **4.8** Abstraction- abstract class

9. SUGGESTED SPECIFICATION TABLE FOR QUESTIONPAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Basics of Python	07	6	7	5	18

II	Lists, Tuples, Sets, Dictionaries and String	10	9	9	7	25
III	Functions and Modules	05	3	3	4	10
IV	Object Oriented Programming	06	6	6	5	17
Total		**28**	**24**	**25**	**21**	**70**

Legends: *R=Remember, U=Understand, A=Apply and above (Revised Bloom's taxonomy)*
Note*: This specification table provides general guidelines to assist students for their learning and to teachers to teach and question paper designers/setters to formulate test items/questions to assess the attainment of the UOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may slightly vary from above table.*

10. SUGGESTED STUDENT ACTIVITIES

Other than the classroom and laboratory learning, following are the suggested student-related ***co-curricular*** activities which can be undertaken to accelerate the attainment of the various outcomes in this course. Students should perform following activities in group (or individual) and prepare reports of about 5 pages for each activity. They should also collect/record physical evidence for their (student's) portfolio which may be useful for their placement interviews:

a) Undertake micro-projects in teams .
b) Give a seminar on any relevant topics.
c) Participate in online coding challenges and hackathons to improve programming skills in Python.
d) Make a list of the Python-based applications or software.
e) Students are encouraged to register themselves in various MOOCs such as: Swayam, edx, Coursera, Udemy, Sololearn etc. to further enhance their learning.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

These are sample strategies, which the teacher can use to accelerate the attainment of the various outcomes in this course:

a) Massive open online courses (***MOOCs***) may be used to teach various topics/subtopics.
b) Guide student(s) in undertaking micro-projects.
c) ***'L' in section No. 4 means*** different types of teaching methods that are to be employed by teachers to develop the outcomes.
d) Code reviews: Conduct code reviews to provide feedback to students on their coding skills. This will help them to identify areas where they need to improve and also learn best practices from their peers.
e) Online resources: Provide students with access to online resources, such as tutorials, videos, and forums, that will help them to deepen their understanding of Python concepts and also provide them with additional practice opportunities.

12. SUGGESTED MICRO-PROJECTS

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-projects are group-based (group of 3 to 5). However, **in the fifth and sixth semesters**, 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 duration of the micro project should be about **12-14 *(fourteen to sixteen) student engagement hours*** during the course. The students ought to submit 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:

a) Develop a program to simulate a simple game using control structures, functions, and OOP concepts.
b) Create a program to manage a student database using dictionaries, lists, and functions.
c) Develop a program to calculate statistics on a dataset using modules and functions.
d) Implement a simple calculator using OOP concepts, including inheritance and polymorphism.
e) Build a program to analyze text data using string operations, sets, and dictionaries.
f) Create a program to perform image processing using modules and functions.
g) Develop a program to simulate a simple banking system using OOP concepts, including encapsulation and inheritance.
h) Implement a program to manage a simple inventory system using lists and dictionaries.
i) Create a program to generate and manage passwords using string operations and modules.
j) Build a program to implement various sorting algorithms using functions and OOP concepts.
k) Weather Data Visualization: Develop a tool that reads weather data from a file, such as a CSV, and creates graphical representations of the data using libraries such as Matplotlib. (for data visit https://open-meteo.com/, https://www.visualcrossing.com/weather-api etc)
   Weather Data Analytics: Build a program that analyzes large sets of weather data, such as historical temperature, wind speed, and precipitation data, and identifies patterns and trends.
   (for data visit https://open-meteo.com/, https://www.visualcrossing.com/weather-api etc)
l) Air Quality Monitoring: Develop a tool that retrieves air quality data from APIs, stores it in a database, and displays the data using charts and graphs.
   (for data visit https://data.gov.in/catalog/historical-daily-ambient-air-quality-data etc)

13. SUGGESTED LEARNING RESOURCES

Sr. No.	Title of Book	Author	Publication with place, year and ISBN
1	Python Basics: A Practical Introduction to Python 3	David Amos, Dan Bader et. al.	Real Python, 2021 ISBN : 9781775093329
2	Beginning Python	James Payne	Wiley, 2010 ISBN: 9780470414637

3	Python Programming: An Introduction to Computer Science	John Zelle	Franklin, Beedle & Associates Inc; Pap/Cdr edition (1 December 2003) ISBN-10 : 1590280288 ISBN-13 : 978-1887902991
4	Introduction to Problem Solving with Python	E. Balagurusamy	Mc Graw Hill India, New Delhi, 2017 ISBN: 9789352602582
5	Python 3 Object Oriented Programming	Dusty Philips	PACKT Publishing 2010 ISBN 978-1-849511-26-1

14. SOFTWARE/LEARNING WEBSITES

WEBSITE

- https://www.python.org/
- https://www.sololearn.com/
- https://realpython.com/
- https://nptel.ac.in/courses/106106182
- https://nptel.ac.in/courses/106106145
- https://www.w3schools.com/python/
- https://www.youtube.com/watch?v=_uQrJ0TkZlc

15. PO-COMPETENCY-CO MAPPING:

Semester V	**OOPS & Python Programming (Course Code:4351108)**						
	POs						
Competency & Course Outcomes	PO 1 Basic & Discipline specific knowledge	PO 2 Problem Analysis	PO 3 Design / development of solution	PO4 Engineering Tools, Experimen-tation & Testing	PO 5 Engineering practices for society, sustainability & environment	PO 6 Project Manage-ment	PO 7 Life-long learning
Competency	**Develop program using Object-Oriented Python Programming (OOPS) to solve given problem.**						
Course Outcomes CO1 Demonstrate proficiency in Python programming fundamentals, including data types, control structures, loops, input-output functions.	3	2	1	1	-	1	2
CO2 Develop python programs by applying data structures – list, dictionary, tuple, set and strings concepts.	2	3	2	2	-	2	3
CO3 Design Python functions and modules.	2	1	3	2	-	1	3
CO4 Apply object-oriented	2	2	2	2	-	1	2

programming principles using Python.							

*Legend: ‘**3**’ for high, ‘**2**’ for medium, ‘**1**’ for low and ‘**-**’ for no correlation of each CO with PO.*

16. COURSE CURRICULUM DEVELOPMENT COMMITTEE

GTU Resource Persons

S. No.	Name and Designation	Institute	Contact No.	Email
1.	Dr. S N Sampat, HOD	L E College (Diploma)Morbi	-	snsampat@gmail.com
2.	Mr. Ashish M Patel, Lecturer	Government Polytechnic for Girls, Surat	-	gpgsecamp@gmail.com

BoS Resource Persons

Sr. No	Name and Designation	Institute	Contact No.	Email
1.	Dr. A S Pandya, Principal BoS Chairman Electrical & Allied Branches	BPTI, Bhavnagar	9426201171	aspandya22@rediffmail.com
2.	Shri U V Buch, LEC BoS Member & Branch Coordinator-EC	GP A’bad	9825346922	uvbuch@gmail.com