GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)

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

Course Title: Python Programming
(Course Code: C4323203)

Diploma programme in which this course is offered	Semester in which offered
Information Communication Technology	Second

1. 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 Information Communication Technology (ICT) 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.

2. COMPETENCY

The purpose of this course is to enable the student to attain the following industry identified competency through various teaching-learning experiences:

- Develop Programs using Python to solve well-defined Problems.

3. COURSE OUTCOMES (COs)

The practical exercises, the underpinning knowledge, and the relevant soft skills associated with this competency are to be developed in the student to display the following COs:

The student will develop underpinning knowledge, adequate programming skills of competency for implementing various applications using python to attain the following course outcomes.

a) Prepare flowchart and algorithm for solving computing problems.
b) Develop python programs to solve simple problems.
c) Apply control structure feature of python for developing programs.
d) Develop Python Programs using built-in functions, modules and libraries.
e) Develop python programs using dictionary, list, string, set and tuple manipulation functions.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme (In Hours)			Total Credits (L+T/2+P/2)	Examination Scheme				
				Theory Marks		Practical Marks		Total Marks
L	T	P	C	CA	ESE	CA	ESE	
3	0	4	5	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 COs. . These PrOs need to be attained to achieve the COs.

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. required
1	Prepare flowchart and algorithm for a given problem.(Following are the sample programs. Faculty can select any other similar programs for the practice of the students.) i. Find the sum of two given numbers. ii. Find a maximum out of two given numbers. iii. Find whether a given number is odd or even. iv. Find a maximum out of three given numbers.	I	02
2	Install & configure python software.	II	01
3	Write a Program to print your name, mobile number, and date of birth.	II	01

4	Develop a Program to identify data-types in python.	II	01
5	Create a Program to read three numbers from the user and find the average of the numbers.	II	01
6	Develop a Program that can calculate simple interest and compound interest on given data.	II	02
7	Write a Program to convert temperature from Fahrenheit to Celsius unit using eq: C=(F-32)/1.8	II	02
8	Identify whether the scanned number is even or odd and print an appropriate message.	III	02
9	Create a Program to find a maximum number among the given three numbers.	III	02
10	Develop a Program to demonstrate the basic functionalities of a standard calculator.	III	02
11	Write a Python program to print 1 to 10 numbers using loops.	III	02
12	Develop a Program to find odd and even numbers from 1 to N numbers. (Where N is an integer number)	III	02
13	Write a Program to show whether the entered number is prime or not.	III	02
14	Write a Program to display the following patterns. A) 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 B) * * * * * * * * * * * * * * *	III	02
15	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)	IV	02
16	Develop a User-defined function to find the factorial of a given number.	IV	02
17	Write a Program using the function that reverses the entered value.	IV	02

18	Write a Program that determines whether a given number is an 'Armstrong number' or a palindrome using a user-defined function.	IV	02
19	Write a Program to find the length of a string.	V	02
20	Write a Program to reverse words in a given sentence.	V	02
21	Write a Program to check if a substring is present in a given string.	V	02
22	Write a Program to count and display the number of vowels, consonants, uppercase, lowercase characters in a string.	V	02
23	Develop Programs to perform the following list operations. a)To swap given two elements in a list. b)To find the sum of elements in a list.	V	02
24	Develop Programs to perform the following list operations. a)To check if an element exists in a given list. b)To find the smallest and largest element in a given list.	V	02
25	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. b) Create a List containing the square of all odd numbers from range 1 to 10. c) Create a List of prime and non-prime numbers in range 1 to 50.	V	02
26	Write a Program to demonstrate the set functions and operations.	**V**	**02**
27	Write a Program to demonstrate tuples functions and operations	**V**	**02**
28	Write a Program to demonstrate the dictionaries functions and operations.	**V**	**02**
29	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**
30	Write a Program to count the number of times a character appears in a given string using a dictionary.	**V**	**02**
	Total		**56**

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..*

S. 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**

6. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

These major equipment with broad specifications for the PrOs is a guide to procure them by the administrators to usher in uniformity of practicals in all institutions across the state.

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 Open Source : IDLE, Jupyter	2 to 8

7. 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 competency.

a) Work as a leader/a team member.
b) Follow 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.

9. UNDERPINNING THEORY

Only the major Underpinning Theory is formulated as higher-level UOs of *Revised Bloom's taxonomy* in order development of the COs and competency is not missed out by the students and teachers. If required, more such higher-level UOs could be included by the course teacher to focus on the attainment of COs and competency.

Unit	Unit Outcomes (UOs) (4 to 6 UOs at Application and above level)	Topics and Sub-topics
Unit – I Problem Solving using Flowchart and Algorithm	1.1 Explain steps to solve a problem. 1.2 Draw flowchart of given problem based on sequence, selection, repetition. 1.3 Write pseudocode for the given problems.	1.1.1 Introduction, Steps for problemsolving, Algorithm and its characteristics, Importance of flowchart and algorithm 1.2.1 Symbolic representation of a flowchart, Limitations of flowchart Flow of control 1.3.1 Problem solving using pseudocode
Unit – II Python Introduction	2.1 Explain the given features and applications of python. 2.2 Install the latest version of python 2.3 Apply given keywords, identifiers, variables, data types, and operators in python programs. 2.4 Write pseudocode for the explicit and implicit type conversion.	2.1.1 Introduction to python, Python features, Applications of python programming 2.2.1 Python installation 2.3.1 Basic structure of python program, Keywords, identifiers, and variables, Data types, Operators 2.4.1 Type Conversion
Unit– III Flow of Control	3.1 Classify control structure. 3.2 Write pseudocode for the flow of control using different selection types. 3.3 Write pseudocode for the flow of control using different repetition types.	3.1.1 Introduction to Flow of Control 3.2.1 Selection - If statement - Elif statement - Nested if statement 3.3.1 Repetition - For loop

	3.4 Implement break and continue statements in the python program.	- While loop - Nested loop 3.4.1 Break and Continue Statements
Unit– IV Functions	4.1 Write pseudocode for userdefined functions. 4.2 Apply Global and Local variable concepts in python program. 4.3 Use built-in functions and modules.	4.1.1 Introduction to Functions - User Defined Functions - Arguments and Parameters 4.2.1Scope of a Variable - Global Variable - Local Variable 4.3.1 Python Standard Library - Built-in functions - Input or output - input() , print() - Mathematical Functions - abs(), divmod(), max(), min(), pow(), sum() - Module - math - random - statistics

Unit– V Dictionary, List, Set, String and Tuple	5.1 Use of string operations in python programs. 5.2 Use built-in functions for string manipulation. 5.3 Write pseudocode for list operations. 5.4 Use built-in functions for list manipulation. 5.5 Use built-in functions for Set manipulation. 5.6 Use of Tuple operations in python programs. 5.7 Use built-in functions for Dictionary manipulation.	5.1.1 Introduction to String, String Operations, Traversing a String 5.2.1 String Methods and Built-in Functions 5.3.1 Introduction to List and its Operations 5.4.1 List Methods and Built-in Functions 5.5.1 Set ● Create a Set, Accessing Python Sets, Delete from set, Update set ● Python Set Operations 5.6.1 Tuple ● Creating Tuples ● Accessing Tuple -Iterate over tuple and Slicing tuple ● Python Tuple Operations, Functions and Methods 5.7.1 Dictionary ● Creating Dictionary ● Accessing Items in Python Dictionary ● Add, Update, Remove in Dictionary ● Built-In Dictionary Methods and functions

***Note**: The UOs need to be formulated at the 'Application Level' and above of Revised Bloom's Taxonomy' to accelerate the attainment of the COs and the competency.*

10.SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A	Total Marks
I	Problem Solving using Flowchart and Algorithm	04	02	02	04	08
II	Introduction to Python	06	04	04	04	12
III	Flow of Control	10	04	04	08	16
IV	Functions	08	04	04	08	16

V	Dictionary, List, Set, String and Tuple	14	02	06	10	18
Total		**42**	**16**	**20**	**34**	**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 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 vary slightly from the 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 conduct following activities in group and prepare reports of about 5 pages for each activity, also collect/record physical evidences for their (student's) portfolio which will be useful for their placement interviews:

a) Prepare a document which differentiates python versions.
b) Undertake micro-projects in teams
c) Give a seminar on any relevant topics.
d) Undertake a market survey of different python frameworks.

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) About ***20% of the topics/sub-topics*** which are relatively simpler or descriptive in nature is to be given to the students for ***self-learning***, but to be assessed using different assessment methods.
e) With respect to ***section No.11***, teachers need to ensure to create opportunities and provisions for ***co-curricular activities***.
f) Guide students for open source python editors.

13. 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-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, workshopbased, 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 ICT 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

14. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication with place, year and ISBN
1	Learn Programming in Python with Cody Jackson	Cody Jackson	Packt Publishing, 2018, ISBN : 9781789531947
2	Python Basics: A Practical Introduction to Python 3	David Amos, Dan Bader et. al.	Real Python, 2021 ISBN : 9781775093329
3	Introduction to Problem Solving with Python	E. Balagurusamy	Mc Graw Hill India, New Delhi, 2017 ISBN: 9789352602582
4	Beginning Python	James Payne	Wiley, 2010 ISBN: 9780470414637
5	Think Python	Allen Downey	O'Reilly, USA, 2016, ISBN : 9789352134755

15. SOFTWARE/LEARNING WEBSITES

a) www.python.org
b) www.learnpython.org
c) www.hackr.io/tutorials/learn-python
d) www.sololearn.com/learning/1073
e) www.nptel.iitm.ac.in

16. PO-COMPETENCY-CO MAPPING

Semester II	Basic Electronics (Course Code: 1323203)									
	POs and PSOs									
Competency & Course Outcomes	**PO 1** Basic & Discipline specific knowledge	**PO 2** Problem Analysis	**PO 3** Design/ development of solutions	**PO 4** Engineering Tools, Experimentation &Testing	**PO 5** Engineering practices for society, sustainability & environment	**PO 6** Project Management	**PO 7** Lifelong learning	**PSO 1**	**PSO 2**	**PSO 3** (If needed)
Competency **Develop Programs using Python to solve well-defined Problems.**										
Course Outcomes CO a) Prepare flowchart and algorithm for solving computing problems.	3	2	3	2	-	2	3			
CO b) Develop python programs to solve simple problems.	3	2	3	2	-	2	3			
CO c) Apply control structures of python for developing programs.	3	-	3	3	-	3	3			
CO d) Develop a program in Python using built-in functions, modules, and library functions.	-	3	3	3	-	3	3			
CO e) Develop python programs using dictionary, list, string, set and tuple manipulation functions.	-	3	3	3	-	3	2			

Legend: '**3**' for high, '**2**' for medium, '**1**' for low or '**-**' for the relevant correlation of each competency, CO, with PO/ PSO

17. COURSE CURRICULUM DEVELOPMENT COMMITTEE

S. No.	Name and Designation	Institute	Contact No.	Email
1	Dr. Parvez Faruki	A.V.P.T.I. Rajkot	0281 248 0175	Parvezfaruki.kg@gmail.com
2	Ms. Ayesha S. Shaikh	R. C. Technical Institute, Ahmedabad	079-27664785-(Ext. 961 & 962)	shaikh.ayesha0014@gmail.com
3	Mr. Mayank V. Parekh	Government	079-26301285	mayank.parekh50@gmail.com

		Polytechnic, Ahmedabad		
4	Mr. Hardik Mandora	R. C. Technical Institute, Ahmedabad	079-27664785 -(Ext. 961 & 962)	hmandora@gmail.com