GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)

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

Course Title: **Advanced Python Programming**
(Course Code: 4321602)

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

1. RATIONALE

In previous semester (Semester I) students have learned basics of python programming. They understand developing programs to solve simple problems using various features of the ‘Python’ language. Python is a modern language useful for writing compact codes specifically for programming in the area of Server-side Web development, Data Analytics, AI, and scientific computing as well as production tools and game programming. This course deals with some advanced features of the ‘Python’ language. The programming skills thus acquired can be used for developing programs with advanced level programming features which in turn will be helping in developing practical applications for the scientific, research and business purposes. At the end of the course, the student will be able to develop programs with adequate advanced concepts using python language.

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 a program using advanced python language concepts to solve the given problem.**

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 programming language to attain the following course outcomes.

a) Develop python programs by applying data structures - dictionary, tuple, and set concepts.
b) Develop modules and packages in python programs for modular programming approach.
c) Implement error handling techniques using exception handling.
d) Develop python programs using file input/output operations.

e) Draw graphics using the turtle module.

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	-	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	Write a program to demonstrate the set functions and operations.	I	02
2	Write a program to input n numbers from the user and store these numbers in a tuple. Print the maximum and minimum number from this tuple.	I	02
3	Create a user-defined function that prints a tuple whose values are the cube of a number between 1 and n (both included), Where n is an integer number and passed as an argument.	I	02
4	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.	I	02
5	Write a program to demonstrate tuples functions and operations	I	02
6	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.	I	02
7	Write a program to count the number of times a character appears	I	02

	in a given string using a dictionary.		
8	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 ‘.	I	02

9	Write a program to concatenate the following dictionaries to create a new one.
	Sample Dictionary:
	dic1={1:10, 2:20}
	dic2={3:30, 4:40}
	dic3={5:50,6:60}
	Expected Result: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
		I	02

10	Write a program to demonstrate the dictionaries functions and operations.	I	02
11	Write a program to define a module to find the area and circumference of a circle. a) import the module to another program. b) import a specific function from a module to another program.	II	02
12	Create a package named DemoPackage which contains two modules named mathematics and greets. The mathematics module contains sum, average, power functions, and the greets module contains the sayHello function. a) import the module from a package to another program. b) import a specific function from a module.	II	02
13	Install urllib3 package using PIP. Send HTTP requests to any URL and print status for the same.	II	02
14	Write a program to catch on Divide by zero Exception with finally clause.	III	02
15	Write a user-defined exception that could be raised when the text entered by a user consists of less than 10 characters.	III	02
16	Write a python program to demonstrate exception handling.	III	02
17	Write a program to read the content of file line by line and write it	IV	02

	to another file except for the lines containing "a" letter in it.		
18	Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.	IV	02
19	Write a program to create a binary file to store Rollno and Name, Search any Rollno and display name if Rollno found otherwise “Rollno not found”.	IV	04
20	Write a program to demonstrate the file and file I/O operations.	IV	02
21	Draw square, rectangle, and circle using Turtle.	V	02
22	Draw color-filled shapes (square, rectangle, and circle) using Turtle.	V	02
23	Draw a smiling face emoji and rainbow using Turtle.	V	04
24	Draw an Indian Flag and an Olympic Symbol using Turtle	V	04
25	Draw a chessboard using Turtle.	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	

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.

8. 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 **Basic of python data structure - Dictionary, Tuple and Set**	1.1 Use of String and list operations in python programs. 1.2 Use built-in functions for Set manipulation. 1.3 Use of Tuple operations in python programs. 1.4 Use built-in functions for Dictionary manipulation.	1.1.1 Introduction to Strings, Lists 1.2.1 Set ● Sets in Python ● Create a Set, Accessing Python Sets, Delete from sets, Update sets ● Python Set Operations 1.3.1 Tuple ● Tuples in Python ● Creating Tuples ● Accessing Tuple ○ Iterate over tuples ○ Slicing tuples ● Tuples are Immutable ● Python Tuple Operations ● Built-In Tuple Functions and methods 1.4.1 Dictionary ● Dictionaries in Python ● Creating Dictionaries ● Accessing Items in Python Dictionaries ● Add, Update, Remove in Dictionaries ● Properties of Dictionary Keys ● Built-In Dictionary Methods and functions
Unit – II **Modules and Packages**	2.1 Describe creating and importing module 2.2 Describe creating and importing package 2.3 PIP - Package Installer for Python	2.1.1 Introduction to module 2.1.2 Creating user defined module 2.1.3 Importing a module in python ● Normal import ● From import ● From import with * 2.1.4 Module search path 2.2.1 Introduction to Packages 2.2.2 Creating user defined package 2.2.3 Importing a package in python ● Normal import

		● From import ● From import with * 2.2.4 Intra-package References 2.3.1 Installing PIP 2.3.2 Installing/uninstalling python packages
Unit– III Exception handling	3.1 Explain errors & exceptions 3.2 List types of Exceptions 3.3 Implement Raising Exceptions 3.4 Implement Handling Exceptions	3.1.1 Introduction to Exception 3.2.1 Types of Exceptions : ● Built-in Exceptions ● User defined Exceptions 3.3.1 Raising Exceptions 3.4.1 Handling Exceptions ● Try clause ● Except clause ● Finally clause
Unit– IV Files Handling	4.1 List types of file 4.2 Explain basics of file creation, reading and writing in context to file handling. 4.3 Explain setting file offsets 4.4 Explain Object Serialization.	4.1.1 Introduction to files and its types. ● Binary files ● Text files 4.2.1 Opening and Closing Text File 4.2.2 Reading and Writing Files 4.3.1 Setting Offsets in File 4.4.1 Object Serialization - Pickle Module
Unit– V Graphics with Turtle	5.1 Explain turtle graphics module 5.2 Implement graphics using turtle 5.3 Use loops and conditional statements to draw graphics	5.1.1 Introduction to turtle graphics 5.2.1 Turtle methods 5.2.2 TurtleScreen Methods 5.3.1 Turtle programming - loops and conditional statements.

***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.*

9. SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Basic of python data structure - Dictionary, Tuple and Set	10	04	06	08	18
II	Modules and Packages	08	02	04	06	12

III	Exception handling	08	02	06	06	14
IV	Files Handling	08	04	04	06	14
V	Graphics with Turtle	08	02	02	08	12
Total		**42**	**14**	**22**	**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.

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-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:** Create an English dictionary that is able to perform the following function.
  i) Add a word and its meaning.
  ii) Delete a word and its meaning.
  iii) Update word or it's meaning.
  iv) Print a list of word and their meaning.
- **Case Study 2:** Build the famous snake game with python's turtle module.
- **Case Study 3:** Draw the famous cartoon character Doraemon, Shinchan with Turtle module

13. 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 : 978-9352134755

14. 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
f) https://docs.python.org/3/library/turtle.html

15. PO-COMPETENCY-CO MAPPING

Semester II	Advanced Python Programming (Course Code: 4321602)									
	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 Life-long learning	PSO 1	PSO 2	PSO 3 (If needed)
Competency **Develop a program using advanced python language concepts to solve the given problem.**										
Course Outcomes CO a) Develop python programs by applying Python data structures - dictionary, tuple and set concepts..	3	2	3	2	-	2	3			
CO b) Develop modules and packages in Python programs for modular programming approach.	3	2	3	2	-	2	3			
CO c) Implement error handling techniques using exception handling.	3	2	3	2	-	3	3			
CO d) Develop python programs using File Input/output operations.	3	2	3	2	-	3	3			
CO e) Draw graphics using turtle module	3	3	3	3	-	3	3			

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

16. COURSE CURRICULUM DEVELOPMENT COMMITTEE

GTU Resource Persons

S. No.	Name and Designation	Institute	Email
1	Prof (Dr.) Parvez Faruki	HOD-I.T. AVPTI,Rajkot	Parvezfaruki.kg@gmail.com
1	Prof. Nandu A. Fatak	HOD-I.T. LEC Poly Morbi	nandu_fatak@yahoo.com
2	Ms. Ayesha S. Shaikh	R.C.Technical Institute, Ahmedabad	shaikh.ayesha0014@gmail.com
3	Mr. Pradipsinh K. Chavda	LEC Poly. Morbi	pradipchavda.it@gmail.com