GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)

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

Course Title: Programming in Python
(Course Code: 4362004)

Diploma program in which this course is offered	Semester in which offered
Mechatronics Engineering	Sixth

1. **RATIONALE**

   Computer programming skills are now becoming part of basic education as these skills are increasing of vital importance for future job and career prospects. The Python programming language is one of the most popular programming languages worldwide. The course emphasizes the use of python programming in multiple domains.

   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.

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 concepts of python to solve the 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 following Course Outcomes (COs) achievement:
   a) Develop python programs to solve simple problems.
   b) Apply control structure feature of python for developing programs.
   c) Develop Python Programs using User-defined, built-in and library functions
   d) Develop Python Programs by applying data structures like strings, lists, dictionary, tuple and set.
   e) Develop modules and packages in python programs for modular programming approach.

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	
3	0	2	4	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 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 sub-components of the COs. These PrOs need to be attained to achieve the COs.

Sr. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. required
1	Install & Configure Python Software and Test a Python Program which prints your name, mobile number, and date of birth.	I	02
2	Create a Python Program which 1) Identifies data-types in python. 2) Converts temperature from Fahrenheit to Celsius unit using equation: C = (F-32)/1.8.	I	02
3	Write a python program to identify whether the scanned number is even or odd and print an appropriate message.	II	02
4	Create a program to find a maximum number among the given three numbers.	II	02
5	Write a python program to print 1 to 10 numbers using for & while loops.	II	02
6	Develop a user-defined function which identifies odd and even numbers from 1 to N numbers.	III	02
7	Write a program 1) To find the length of a string. 2) To check if a substring is present in a given string.	IV	02
8	Develop programs to perform the following list operations. 1) To find the sum of elements in a list. 2) To check if an element exists in a given list.	IV	02

	3) To find the smallest and largest element in a given list.		
9	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.	IV	02
10	Write a program to demonstrate tuple functions and operations	IV	02
11	Write a program to demonstrate the set functions and operations.	IV	02
12	Write a program which defines a module to perform various arithmetic operations.	V	02
13	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. 1) Import the module from a package to another program. 2) Import a specific function from a module.	V	04
Total			**28**

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**

6.   MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

Sr. No.	Equipment Name with Broad Specifications	PrO. No.
1	Computer system with operating system: Windows 7 or higher Ver., Mac OS, and Linux, with 4GB or higher RAM, Python versions: 2.7.X, 3.6.X	All

Sr. No.	Equipment Name with Broad Specifications	PrO. No.
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 course competency.

a)   Follow safety practices.

b)   Practice good housekeeping.

c)   Demonstrate working as a leader/a team member.

d)   Maintain tools and equipment

e)   Follow ethical practices.

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)	Topics and Sub-Topics
Unit – I Introduction to Python	1.1 Explain the given features and applications of python. 1.2 Write steps to Install the latest version of python 1.3 Apply given keywords, identifiers, variables, data types, and operators in python programs. 1.4 Write python program for the explicit and implicit type conversion.	1.1.1 Introduction to python, Python features, Applications of python programming 1.2.1 Python installation 1.3.1 Basic structure of python program, Keywords, identifiers, and variables, Data types, Operators 1.4.1 Type Conversion
Unit – II Flow of Control	2.1 Classify control structure. 2.2 Write python program for the flow of control using different selection types. 2.3 Write python program for the flow of control using different repetition types. 2.4 Implement break and continue statements in the python program.	2.1.1 Introduction to Flow of Control 2.2.1 Selection • If statement • Elif statement • Nested if statement 2.3.1 Repetition • For loop • While loop • Nested loop 2.4.1 Break and Continue Statements
Unit – III Functions in Python	3.1 Write python program for user-defined functions. 3.2 Apply Global and Local variable concepts in python program. 3.3 Use built-in functions and modules.	3.1.1 Introduction to Functions • User Defined Functions • Arguments and Parameters 3.2.1 Scope of a Variable • Global Variable • Local Variable 3.3.1 Python Standard Library • Built-in functions: input(), print() • Mathematical Functions: abs(), divmod(), max(), min(),

		pow(), sum() • Modules: math, statistics
Unit – IV **Python Data Structures:** **String, List, Dictionary, Tuple and Set**	4.1 Use of string operations in python programs. 4.2 Use of list operations in python programs. 4.3 Use built-in functions for Dictionary manipulation. 4.4 Use of Tuple operations in python programs. 4.5 Use built-in functions for Set manipulation.	4.1.1 Introduction to Strings, String Operations, Strings Methods and Built-in Functions 4.2.1 Introduction to List, List Operations, List Methods and Built-in Functions 4.3.1 Introduction to Dictionary, Dictionary Operations, Dictionary Methods and Built-in Functions 4.4.1 Introduction to Tuple, Tuple Operations, Tuple Methods and Built-in Functions 4.5.1 Introduction to Set, Set Operations, Set Methods and Built-in Functions
Unit – V **Modules and Packages**	5.1 Describe creating and importing module 5.2 Describe creating and importing package 5.3 PIP - Package Installer for Python	5.1.1 Introduction to module 5.1.2 Creating user defined module 5.1.3 Importing a module in python • Normal import • From import • From import with * 5.2.1 Introduction to Packages 5.2.2 Creating user defined package 5.2.3 Importing a package in python • Normal import • From import • From import with * 5.3.1 Installing/uninstalling python packages using PIP

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 QUESTIONPAPER DESIGN:

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total
1	Introduction to Python	04	02	02	04	10
2	Flow of Controls	08	02	02	08	14
3	Functions in Python	12	02	04	08	16
4	Python Data Structures – String, List, Dictionary, Tuple and Set	12	04	04	08	18
5	Modules and Packages	06	02	04	04	12
	Total	**42**	**14**	**20**	**36**	**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 perform following activities in group and prepare reports of about 5 pages for each activity. They should also collect/record physical evidences for their (student's) portfolio which may be useful for their placement interviews:
a) Students are encouraged to learn Visual Language programming like scratch, snap etc.
b) Undertake micro-projects in teams.
c) Prepare charts to explain use/process of the identified topic.
d) https://www.codechef.com/ , in this website very elementary programs are available, students are expected to solve those programs
e) Students are encouraged to register themselves in various MOOCs such as: Swayam, edx, Coursera, Udemy etc to further enhance their learning.
f) Encourage students to participate in different coding competitions like
hackathon, online competitions on codechef etc.

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/sub topics.
b) Guide student(s) to take micro-projects.
c) Blend the basic concepts with more specialized instruction

d) Visualization, Cooperative Learning, inquiry based instruction, differentiation, effective use of technology, think-pair and share etc pedagogies can be implemented as per the enlisted course outcomes.

e) Give at least 10 competitive problems for each course outcomes of this course

f) Practice, practice and practice - expose students to wide range of problems

g) 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.

h) With respect to section No.10, teachers need to ensure to create opportunities and provisions for co-curricular activities.

i) Guide students on how to address issues on environment and sustainability using the knowledge of this course.

12. 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:

1. To-Do List Application: Create a simple to-do list application where users can add, delete, and mark tasks as done.

2. Password Generator: Develop a password generator that creates strong, random passwords with specified criteria.

3. Hangman Game: Implement the classic hangman game where users guess a word letter by letter.

4. Expense Tracker: Develop an application to track and categorize expenses, showing a summary of spending over time.

Additionally Mechatronics Engineering Student can also choose **IoT based Micro Projects** in which Python Fundamentals can be applied. Following is the suggested list.

1. Arduino Communication: Interface an Arduino with Python to read sensor data (e.g., temperature, humidity) and control actuators (e.g., motors, LEDs) through Python scripts.

2. Voice-Controlled Home Automation: Implement a home automation system where devices can be controlled using voice commands. Python can be used for speech recognition and controlling the connected devices.

3. Biometric Door Lock System: Create a door lock system that uses biometric data (e.g., fingerprint) for authentication. Python can be used for interfacing with the biometric sensor and controlling the locking mechanism.

4. Smart Mirror: Build a smart mirror that displays useful information such as time, weather, and calendar events. Python can be used to fetch and process the data.

13. SUGGESTED LEARNING RESOURCES

Sr. 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
4	Beginning Python	James Payne	Wiley, 2010 ISBN: 9780470414637
5	Think Python	Allen Downey	O'Reilly, USA,

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 VI	Programming in Python (Course Code:4362004)						
	POs						
Competency & Course Outcomes	PO 1 Basic & Discipline specific knowledge	PO 2 Problem Analysis	PO 3 Design/ development of solutions	PO 4 Engineering Tools, Experimentation &Testi	PO 5 Engineering practices for society,	PO 6 Project Management	PO 7 Life-long learning

				ng	sustai nabilit y & enviro nmen t		
Competency	**Develop a program using concepts of python language to solve the given problem.**						
Course Outcomes CO1 Develop python programs to solve simple problems.	3	2	3	2	-	2	3
CO2 Apply control structure feature of python for developing programs.	3	-	3	3	-	3	3
CO3 Develop Python Programs using User-defined, built-in and library functions	-	3	3	3	-	3	3
CO4 Develop Python Programs by applying data structures like strings, lists, dictionary, tuple and set.	3	2	3	2	-	2	3
CO5 Develop modules and packages in python programs for modular programming approach.	3	2	3	2	-	2	3

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

Sr. No.	Name and Designation	Institute	Contact No.	Email
1.	Prof. Hitesh B Patel (IT Department)	B & B Institute of Technology, VV Nagar	9601451273	hbpatel@bbit.ac.in

17. BOS Resource Persons

Sr. No.	Name and Designation	Institute	Contact No.	Email
1.	Dr. S. H. Sundarani, BOS Chairman & HOD Mechanical	Government Polytechnic, Ahmadabad	9227200147	gpasiraj@gmail.com
2.	Dr. Rakesh D. Patel, BOS Member & HOD Mechanical	B. & B. Institute of Technology, V. V. Nagar	9825523982	rakeshgtu@gmail.com
3	Dr. Atul S. Shah, BOS Member & Principal	B. V. Patel Institute of Technology, Bardoli	7567421337	asshah97@yahoo.in