Program Name : **Computer Engineering Program Group**

Program Code : **CO/CM/IF/CW**

Semester : **Sixth**

Course Title : **Programming with 'Python'**

Course Code : **22616**

1. RATIONALE

Python is powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python code is simple, short, readable, intuitive, and powerful, and thus it is effective for introducing computing and problem solving to beginners. It's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

2. COMPETENCY

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

- **Develop general purpose programming using python to solve problems**

3. COURSE OUTCOMES (COs)

The theory, practical experiences and relevant soft skills associated with this course are to be taught and implemented, so that the student demonstrates the following ***industry-oriented*** COs associated with the above-mentioned competency:

a) Display message on screen using Python script on IDE.
b) Develop python program to demonstrate use of Operators
c) Perform operations on data structures in Python.
d) Develop functions for given problem.
e) Design classes for given problem.
f) Handle exceptions.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme			Credit (L+T+P)	Examination Scheme												
L	T	P		Theory							Practical					
				Paper Hrs.	ESE		PA		Total		ESE		PA		Total	
					Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
3	-	2	5	3	70	28	30*	00	100	40	25@	10	25	10	50	20

(*): Under the theory PA, out of 30 marks, 10 marks are for micro-project assessment to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessment of the UOs required for the attainment of the COs.

***Legends: L**-Lecture; **T** – Tutorial/Teacher Guided Theory Practice; **P** - Practical; **C** – Credit, **ESE** - End Semester Examination; **PA** - Progressive Assessment*

5. COURSE MAP (with sample COs, PrOs, UOs, ADOs and topics)

This course map illustrates an overview of the flow and linkages of the topics at various levels of outcomes (details in subsequent sections) to be attained by the student by the end of the course, in all domains of learning in terms of the industry/employer identified competency depicted at the centre of this map.

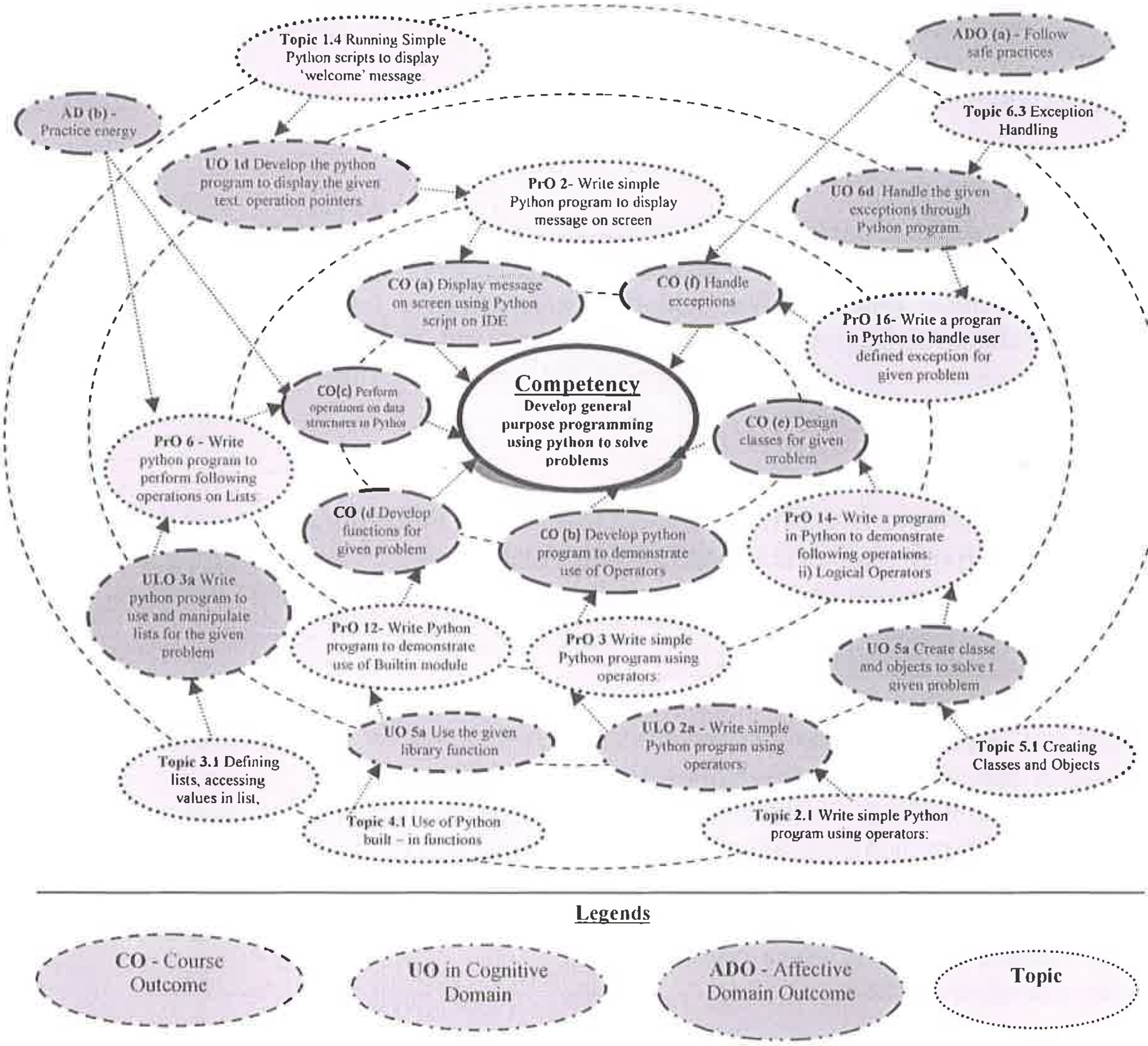
Topic 1.4 Running Simple Python scripts to display 'welcome' message
ADO (a) - Follow safe practices
AD (b) - Practice energy
Topic 6.3 Exception Handling
UO 1d Develop the python program to display the given text. operation pointers
PrO 2- Write simple Python program to display message on screen
UO 6d Handle the given exceptions through Python program
CO (a) Display message on screen using Python script on IDE
CO (f) Handle exceptions
PrO 16- Write a progran in Python to handle user defined exception for given problem
CO(c) Perform operations on data structures in Pythor
Competency
Develop general purpose programming using python to solve problems
CO (e) Design classes for given problem
PrO 6 - Write python program to perform following operations on Lists
CO (d Develop functions for given problem
CO (b) Develop python program to demonstrate use of Operators
PrO 14- Write a program in Python to demonstrate following operations: ii) Logical Operators
ULO 3a Write python program to use and manipulate lists for the given problem
PrO 12- Write Python program to demonstrate use of Builtin module
PrO 3 Write simple Python program using operators:
UO 5a Create classe and objects to solve t given problem
UO 5a Use the given library function
ULO 2a - Write simple Python program using operators.
Topic 3.1 Defining lists, accessing values in list,
Topic 5.1 Creating Classes and Objects
Topic 4.1 Use of Python built – in functions
Topic 2.1 Write simple Python program using operators:
Legends
CO - Course Outcome
UO in Cognitive Domain
ADO - Affective Domain Outcome
Topic

Figure 1 - Course Map

6. SUGGESTED PRACTICALS/ EXERCISES

The practicals in this section are PrOs (i.e. sub-components of the COs) to be developed and assessed in the student for the attainment of the competency.

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
1	Install and configure Python IDE	I	02
2	Write simple Python program to display message on screen	I	02
3	Write simple Python program using operators: a) Arithmetic Operators b) Logical Operators c) Bitwise Operators	II	02
4	Write simple Python program to demonstrate use of conditional statements: a) 'if' statement	II	02

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
	b) ‘if … else’ statement c) Nested ‘if’ statement		
5	Write python program to demonstrate use of looping statements: a) ‘while’ loop b) ‘for’ loop c) Nested loops	II	02
6	Write python program to perform following operations on Lists: a) Create list b) Access list c) Update list (Add item, Remove item) d) Delete list	III	02
7	Write python program to perform following operations on Tuples: a) Create Tuple b) Access Tuple c) Update Tuple d) Delete Tuple	III	02
8	Write python program to perform following operations on Tuples: a) Create Set b) Access Set elements c) Update Set d) Delete Set	III	02
9	Write python program to perform following operations on Dictionaries: a) Create Dictionary b) Access Dictionary elements c) Update Dictionary d) Delete Set e) Looping through Dictionary	III	02
10	a) Write Python program to demonstrate math built- in functions (Any 2 programs) b) Write Python program to demonstrate string built – in functions (Any 2 programs)	IV	02
11	Develop user defined Python function for given problem: a) Function with minimum 2 arguments b) Function returning values	IV	02
12	Write Python program to demonstrate use of: a) Builtin module (e.g. keyword, math, number, operator) b) user defined module.	IV	02
13	Write Python program to demonstrate use of: a) built-in packages (e.g. NumPy, Pandas) b) user defined packages	IV	02
14	Write a program in Python to demonstrate following operations: a) Method overloading b) Method overriding	V	02
15	Write a program in Python to demonstrate following operations: a) Simple inheritance b) Multiple inheritance	V	02

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
16	Write a program in Python to handle user defined exception for given problem	VI	02
	Total		**32**

Note

i. *A suggestive list of PrOs is given in the above table. More such PrOs can be added to attain the COs and competency. All the above listed practical need to be performed compulsorily, so that the student reaches the 'Applying Level' of Blooms's 'Cognitive Domain Taxonomy' as generally required by the industry.*

ii. *The 'Process' and 'Product' related skills associated with each PrO are to be assessed according to a suggested sample given below:*

S. No.	Performance Indicators	Weightage in %
1	Correctness of business logic	40
2	Debugging ability	20
3	Quality of input and output displayed (messaging and formatting)	10
4	Answer to sample questions	20
5	On time term work submission	10
	Total	**100**

The above PrOs also comprise of the following social skills/attitudes which are Affective Domain Outcomes (ADOs) that are best developed through the laboratory/field based experiences:

a) Work collaboratively in team
b) Follow ethical practices.

The ADOs are not specific to any one PrO, but are embedded in many PrOs. Hence, the acquisition of the ADOs takes place gradually in the student when s/he undertakes a series of practical experiences over a period of time. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

- 'Valuing Level' in 1st year.
- 'Organization Level' in 2nd year.
- 'Characterization Level' in 3rd year.

7. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

The major equipment with broad specification mentioned here will usher in uniformity in conduct of practicals, as well as aid to procure equipment by authorities concerned.

S. No.	Equipment Name with Broad Specifications	PrO. S. No.
1	Computer system (Any computer system with basic configuration)	All
2	'Python' Interpreter/ IDE	

8. UNDERPINNING THEORY COMPONENTS

The following topics/subtopics should be taught and assessed to develop UOs in cognitive domain for achieving the COs to attain the identified competency. More UOs could be added.

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
Unit – I Introduction and Syntax of Python Program	1a. Identify the given Variables, Keywords and constants in Python 1b. Use indentation, comments in the given program. 1c. Install the given Python IDE and editor. 1d. Develop the python program to display the given text.	1.1 Features of Python – Interactive, Object – oriented, Interpreted, platform independent 1.2 Python building blocks – Identifiers, Keywords, Indention, Variables, Comments 1.3 Python environment setup – Installation and working of IDE 1.4 Running Simple Python scripts to display 'welcome' message. 1.5 Python Data Types: Numbers, String, Tuples, Lists, Dictionary. Declaration and use of data types
Unit– II Python Operators and Control Flow statements	2a. Write simple Python program for the given arithmetic expressions. 2b. Use different types of operators for writing the the arithmetic expressions. 2c. Write a 'Python' program using decision making structure for two-way branching to solve the given problem. 2d. Write a 'Python' program using decision making structure for multi-way branching to solve the given problem.	2.1 Basic Operators: Arithmetic, Comparison/ Relational, Assignment, Logical, Bitwise, Membership, Identity operators, Python Operator Precedence 2.2 Control Flow: 2.3 Conditional Statements (if, if … else, nested if) 2.4 Looping in python (while loop, for loop, nested loops) 2.5 loop manipulation using continue, pass, break, else.

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
Unit– III Data Structures in Python	3a. Write python program to use and manipulate lists for the given problem 3b. Write python program to use and manipulate Tuples for the given problem 3c. Write python program to use and manipulate Sets for the given problem 3d. Write python program to use and manipulate Dictionaries for the given problem	3.1 Lists: a) Defining lists, accessing values in list, deleting values in list, updating lists. b) Basic List Operations c) Built – in List functions 3.2 Tuples: a) Accessing values in Tuples, deleting values in Tuples, and updating Tuples. b) Basic Tuple operations. c) Built – in Tuple functions 3.3 Sets: a) Accessing values in Set, deleting values in Set and updating Sets. b) Basic Set operations. c) Built – in Set functions 3.4 Dictionaries: a) Accessing values in Dictionary, deleting values in Dictionary and updating Dictionary. b) Basic Dictionary operations. c) Built – in Dictionaries functions
Unit-IV Python Functions, modules, and Packages	4a. Use the Python standard functions for the given problem. 4b. Develop relevant user defined functions for the given problem using Python code. 4c. Write Python module for the given problem 4d. Write Python package for the given problem	4.1 Use of Python built – in functions (e.g. type/ data conversion functions, math functions etc.) 4.2 User defined functions: Function definition, Function calling, function arguments and parameter passing, Return statement, Scope of Variables: Global variable and Local Variable. 4.3 Modules: Writing modules, importing modules, importing objects from modules, Python built – in modules (e.g. Numeric and mathematical module, Functional Programming Module) Namespace and Scoping. 4.4 Python Packages: Introduction, Writing Python packages, Using standard (e.g. math, scipy, Numpy, matplotlib, pandas etc.) and user defined packages
Unit-V Object Oriented Programming in Python	5a Create classes and objects to solve the given problem. 5b Write Python code for data hiding for the given problem. 5c Write Python code using data abstraction for the given problem. 5d Write Python program using inheritance for the	5.1 Creating Classes and Objects. 5.2 Method Overloading and Overriding. 5.3 Data Hiding. 5.4 Data abstraction. 5.5 Inheritance and composition classes 5.6 Customization via inheritance specializing inherited methods.

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
	given problem.	
Unit –VI File I/O Handling and Exception Handling	6a Write Python code for the given reading values from keyboard 6b Read data from the given file. 6c Write the given data to a file. 6d Handle the given exceptions through Python program.	6.1 I/O Operations: Reading keyboard input, Printing to screen 6.2 File Handling: Opening file in different modes, accessing file contents using standard library functions, Reading and writing files, closing a file, Renaming and deleting files, Directories in Python, File and directory related standard functions 6.3 Exception Handling: Introduction, Exception handling - 'try: except:' statement, 'raise' statement, User defined exceptions

***Note**: To attain the COs and competency, above listed UOs need to be undertaken to achieve the 'Application Level' of Bloom's 'Cognitive Domain Taxonomy'*

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	Introduction and Syntax of Python Program	04	02	02	04	08
II	Python Operators and Control Flow statements	06	02	04	04	10
III	Data Structures in Python	12	02	04	08	14
IV	Python Functions, modules, and Packages	12	02	02	10	14
V	Object Oriented Programming in Python	08	02	02	08	12
VI	File I/O Handling and Exception Handling	06	02	02	08	12
Total		**48**	**12**	**16**	**42**	**70**

***Legends:** R=Remember, U=Understand, A=Apply and above (Bloom's Revised taxonomy)*
***Note**: This specification table provides general guidelines to assist student for their learning and to teachers to teach and assess students with respect to attainment of UOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may 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 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 journal of practicals.
b) Undertake micro-projects.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

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

a) Massive open online courses (***MOOCs***) may be used to teach various topics/sub topics.
b) ***'L' in item No. 4*** does not mean only the traditional lecture method, but different types of teaching methods and media that are to be employed to develop the outcomes.
c) About ***15-20% of the topics/sub-topics*** which is relatively simpler or descriptive in nature is to be given to the students for ***self-directed learning*** and assess the development of the COs through classroom presentations (see implementation guideline for details).
d) With respect to item No.10, teachers need to ensure to create opportunities and provisions for ***co-curricular activities***.
e) Guide student(s) in undertaking micro-projects.
f) Demonstrate students thoroughly before they start doing the practice.
g) Encourage students to refer different websites to have deeper understanding of the subject.
h) Observe continuously and monitor the performance of students in Lab.

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 dated work diary consisting of individual contribution 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 micro-project by the end of the semester to develop the industry oriented COs.

A suggestive list of micro-projects is given here. Similar micro-projects could be added by the concerned faculty:

a) Create an English dictionary which is able to perform following function.
   i. Add a word and its meaning.
   ii. Delete a word and its meaning.
   iii. Update word or its meaning.
   iv. Print list of word and its meaning.
b) To create simple calculator using classes and objects.
c) Develop student management system which will able to:
   i) Add ii) Delete iii) Update iv) Display student related information like Roll No, Name, Age, Address, Email-Id, Contact Number etc.
d) Any other micro-projects suggested by subject faculty on similar line.

(Use functions, Classes, Objects and other features of 'Python' to develop above listed applications)

13. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication
1	Python Programing	Rao, K. Nageswara Shaikh Akbar	Scitech Publications (India) Pvt. Ltd. ISBN: 9789385983450
2	Learning Python	Lutz, Mark	5th Edition, O'Reilly Publication ISBN-13: 978-1449355739
3	Python Essential Reference	Beazley, David	4th Edition, Addison-Wesley Professional, ISBN: 9780672329784
4	Head First Python, 2nd Edition	Paul, Barry	O'Reilly Publication, 2nd Edition, ISBN: 1491919531

14. SOFTWARE/LEARNING WEBSITES

a) https://www.tutorialspoint.com/python/index.htm
b) nptel.ac.in/courses/117106113/34
c) https://www.w3schools.com/python/default.asp
d) https://www.programiz.com/python-programming
e) http://spoken-tutorial.org/
f) https://docs.python.org/3/tutorial/errors.html
g) https://www.w3resource.com/python-exercises/
h) https://www.anandology.com/python-practice-book/