

**Program Name** : Diploma in Artificial Intelligence and Machine Learning  
**Program Code** : AN  
**Semester** : Third  
**Course Title** : Data Structure using Python  
**Course Code** : 22395

### 1. RATIONALE

Python is powerful programming language Python code is simple, short, readable, intuitive, and powerful, and thus it is effective for introducing computing and problem solving to beginners. Data structures are mathematical and logical model of storing and organizing data in a particular way in computer. Python has efficient high-level data structures and a simple but effective approach to object-oriented programming After studying this course, student will be able to understand and identify different types of data structures to solve real life problems.

### 2. COMPETENCY

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

- Implement data structures 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:

- Develop python program using basic syntactical constructs.
- Perform operations on data structures in Python.
- Implement modules, packages in python for given problem.
- Design classes for given problem.
- Implement programs for Arrays and Linked List.
- Develop a program to implement Graphs, Trees data structure.

### 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

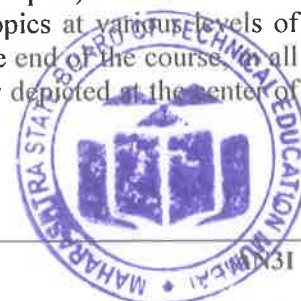
(\*): Under the theory PA, out of 30 marks 10 marks are for micro-project assessment to facilitate attainment of COs and the remaining 20 marks is the average of 2 test to be taken during the semester for the assessment of the cognitive domain UOs required for the attainment of the COs.

#: External Exam

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

### 5. COURSE MAP (with sample COs, Learning Outcomes i.e. LOs 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 center of this map.



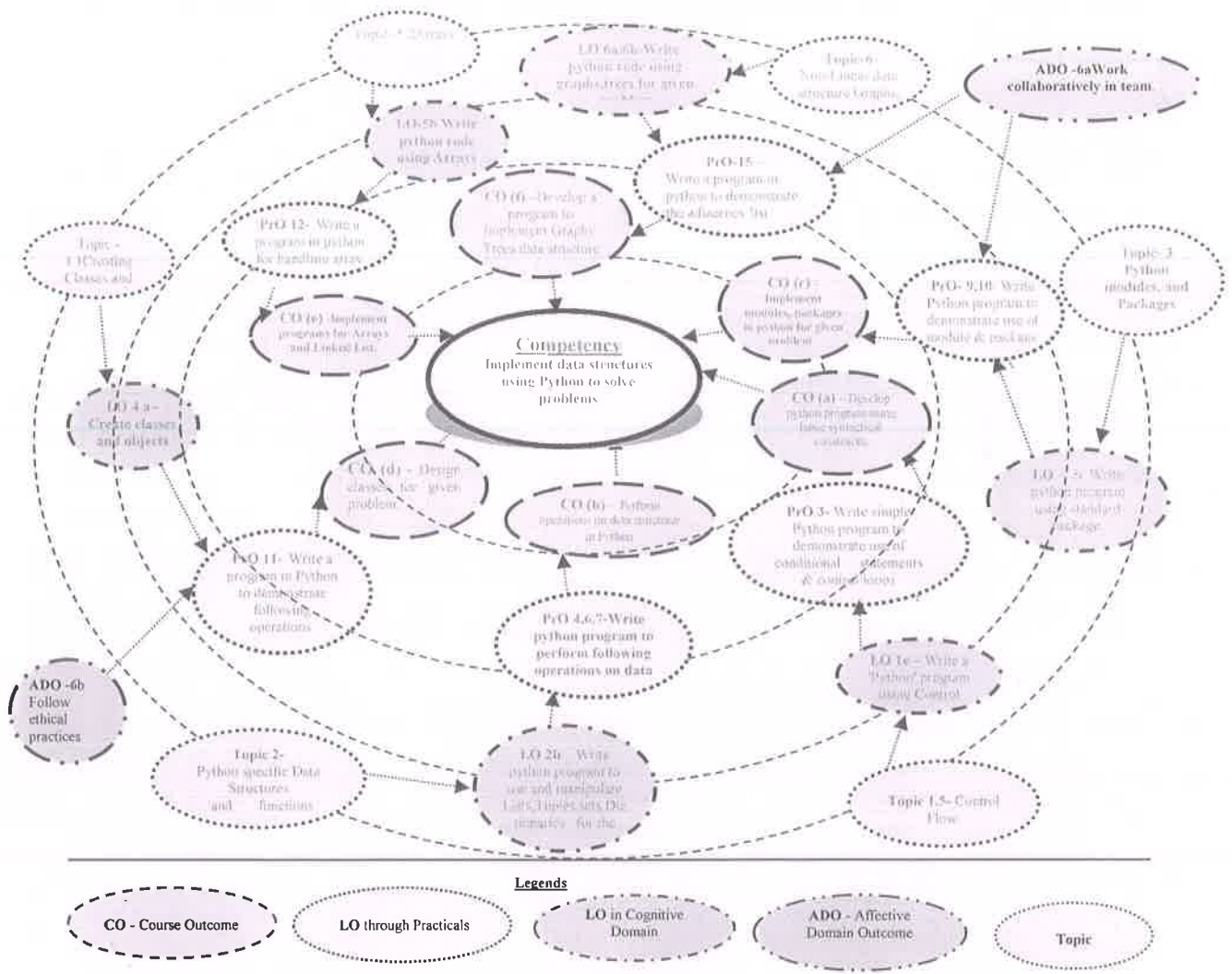


Figure 1 - Course Map

6. SUGGESTED PRACTICALS/ EXERCISES

The practicals/exercises/tutorials in this section are psychomotor domain LOs (i.e. sub-components of the COs) are to be developed and assessed in the student to lead to the attainment of the competency.

Sr. No.	Practical Exercises (Learning Outcomes to be achieved through practicals)	Unit No.	Approx. Hrs. Required
1	Install and configure Python IDE and Write Python program to display message on screen	I	02*
2	Write simple Python program using operators: a. Arithmetic Operators b. Logical Operators c. Bitwise Operators	I	02
3	Write simple Python program to demonstrate use of conditional statements & control loops: a. if b. if...else c. Nested if d. Switch case e. for f. while g. do..... while	I	02*



Sr. No.	Practical Exercises (Learning Outcomes to be achieved through practicals)	Unit No.	Approx. Hrs. Required
4	Write python program to perform following operations on Lists: a. Create b. Access c. Update d. Delete elements in list	II	02*
5	Write python program to perform following operations on Tuples: a. Create b. Access c. Update d. Delete Tuple elements	II	02*
6	Write python program to perform following operations on Set: a. Create b. Access c. Update d. Delete Access Set elements	III	02
7	Write python program to perform following operations on Dictionaries: a. Create b. Access c. Update d. Delete e. Looping through Dictionary	III	02*
8	a. Write Python program to demonstrate math built- in functions b. Write Python program to demonstrate string built - in functions c. Develop user defined Python function for given problem: - Function with minimum 2 arguments	III	02*
9	Write Python program to demonstrate use of: a. Built-tin module (e.g. keyword, math, number, operator) b. User defined module.	III	02*
10	Write Python program to demonstrate use of: a. Built-in packages (e.g. NumPy, Pandas, matplotlib) b. User defined packages	IV	02*
11	Write a program in Python to demonstrate following operations: a. Method overloading b. Method overriding c. Single inheritance	IV	02*
12	Write a program in python for handling array to demonstrate following operations a. Array declaration b. Insertion c. Deletion d. Append e. Index f. Reverse	V	02*
13	Write a program in python for linked list to demonstrate following operations a) Insertion b) Deletion c) Updating d) Merging to list	V	02*



Sr. No.	Practical Exercises (Learning Outcomes to be achieved through practicals)	Unit No.	Approx. Hrs. Required
14	Write a program in python to demonstrate queues using list, deque	V	02*
15	Write a program in python to demonstrate the adjacency list representation of graphs	VI	02*
16	Write a program in python to Create binary tree from the given list using Binary tree module in python	VI	02*
<b>Total</b>			<b>32</b>

**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. A judicious mix of minimum 12 or more practical need to be performed, out of which, the practicals marked as '\*' are compulsory, so that the student reaches the 'Precision Level' of Dave's 'Psychomotor Domain Taxonomy' as generally required by the industry.
- ii. The 'Process' and 'Product' related skills associated with each PrO is 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
<b>Total</b>		<b>100</b>

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 1<sup>st</sup> year.
- 'Organization Level' in 2<sup>nd</sup> year.
- 'Characterization Level' in 3<sup>rd</sup> 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	Expt. 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 in order to develop LOs in cognitive domain for achieving the COs to attain the identified competency.





Unit	Major Learning Outcomes (in cognitive domain)	Topics and Sub-topics
<b>Unit-I</b> <b>Introduction on and Control Flow statements</b>	1a Install the given Python IDE and editor. 1b Identify the given Variables, Keywords and constants in Python. 1c Use indentation, comments in the given program. 1d. Use different types of operators for writing arithmetic expressions. 1e. Write a 'Python' program using Control flow.	1.1 Features of Python - Interactive, Object oriented, Interpreted, platform independent 1.2 Python building blocks - Identifiers, Keywords, Indention, Variables, Comments 1.3 Python Data Types: Numbers, String, Tuples, Lists, Dictionary. 1.4 Basic Operators: Arithmetic, Comparison/ Relational, Assignment, Logical, Bitwise, Membership, Identity operators , Python Operator Precedence 1.5 Control Flow: Conditional Statements (if, if else, nested if). Looping in python (while loop, for loop, nested loops) loop manipulation using continue, pass, break, else.
<b>Unit-II</b> <b>Python specific Data Structures and functions</b>	2a. Write python program to use and manipulate Lists for the given problem. 2b. Write python program to use and manipulate Tuples for the given problem. 2c. Write python program to use and manipulate Sets for the given problem. 2d. Write python program to use and manipulate Dictionaries for the given problem. 2e Develop relevant user defined functions for the given problem using Pythoncode.	2.1 Lists: a) Defining lists, accessing values in list, deleting values in list, updating lists. b) Basic List Operations c) Built - in List functions 2.2 Tuples: a) Accessing values in Tuples, deleting values in Tuples, and updating Tuples. b) Basic Tuple operations. c) Built - in Tuple functions 2.3 Sets: a) Accessing values in Set, deleting values in Set and updating Sets. b) Basic Set operations. c) Built - in Set functions 2.3 Dictionaries: a) Accessing values in Dictionary, deleting values in Dictionary and updating Dictionary. b) Basic Dictionary operations. c) Built- in Dictionaries functions 2.4 Use of Python built- in functions (e.g.type/ data conversion functions, math functions etc.) 2.5 User defined functions: Function definition, Function calling, function argument and parameter passing, Return statement, scope of Variables, Global variable and Local Variable



Unit	Major Learning Outcomes (in cognitive domain)	Topics and Sub-topics
<b>Unit-III Python modules, and Packages</b>	3a. Write Python module for the given problem. 3b. Write Python package for the given problem. 3c. Write python program using standard package (NumPy, Pandas).	3.1 Modules: Writing modules, importing modules, importing objects from modules, Python built-in modules (e.g. Numeric and mathematical module, Functional Programming Module) 3.2 Python Packages: Introduction, Writing Python packages, 3.3 Using standard ( scipy, Numpy, matplotlib, pandas). Numpy: Methods in Numpy, Creating arrays and initializing, Reading arrays from files. Special initializing functions, Slicing and indexing, reshaping arrays, combining arrays, NumPy maths. 3.4 Pandas: Methods, Creating Data structures (Series and Data Frames)
<b>Unit-IV Object Oriented Programming in Python</b>	4a. Create classes and objects 4b. Describe Method Overloading , Method Overriding 4c. Write Python code for data hiding for the given problem 4d. Write Python code using data abstraction for the given problem. 4e. Write Python program using inheritance for the given problem	4.1 Creating Classes and Objects. 4.2 Method Overloading and Overriding. 4.3 Data Hiding. 4.4 Data abstraction. 4.5 Inheritance and composition classes Customization via inheritance specializing inherited methods
<b>Unit-V Linear data structure Arrays, Link List, Queues using Python</b>	5a. Describe data structures in python. 5b. Write python code using Arrays for given problem. 5c. Write python code using link list for given problem. 5d. Write python program using queues.	5.1 Data Structures – Definition, Linear Data Structures, Non-Linear Data Structures use of Python Specific Data Structures: List, Tuples, Set, Dictionaries, 5.2 Arrays - Overview, Types of Arrays, Operations on Arrays, Arrays vs List. Time complexity, Searching -Linear Search and Binary Search. Sorting - Bubble Sort, Insertion Sort, 5.3 Linked Lists – Singly Linked Lists, Doubly Linked Lists, Circular Linked Lists. implementation using python packages for link list 5.4 Queues: Implementation of Queue (List & Linked list),
<b>Unit-VI Non-Linear data structure Graphs, Trees using Python</b>	6a. Write python program using graphs 6b. Write python code using trees for given problem	6.1 Graphs -Introduction, Directed vs Undirected Graphs, Weighted vs Unweighted Graphs, Representations, Breadth First Search, Depth First Search. 6.2 Trees - Tree Terminology, Binary Trees: implementation, Tree Traversals, Binary Search Trees

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



### 9. SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER (INTERNAL DESIGN)

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Introduction and Control Flow statements	06	02	02	04	08
II	Python specific Data Structures and functions	08	02	04	06	12
III	Python modules, and Packages	08	02	02	08	12
IV	Object Oriented Programming in Python	06	02	02	04	08
V	Linear data structure Arrays, Link List, Queues using Python	12	02	04	12	18
VI	Non-Linear data structure Graphs, Trees using Python	08	02	02	08	12
<b>Total</b>		<b>48</b>	<b>12</b>	<b>16</b>	<b>42</b>	<b>70</b>

**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 LOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may vary from above table. This specification table also provides a general guideline for teachers to frame internal end semester practical theory exam paper which students have to undertake.

### 10. SUGGESTED STUDENT ACTIVITIES

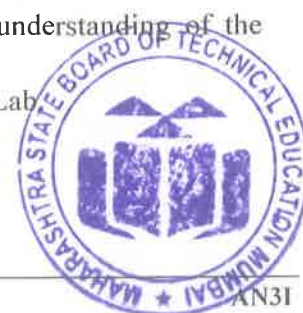
Other than the classroom and laboratory learning, following are the suggested student-related **co-curricular** activities which can be undertaken to accelerate the attainment of the various outcomes in this course: Students should conduct following activities in group and prepare reports of about 5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

- a. Prepare journal of practical's.
- 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 being **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**.

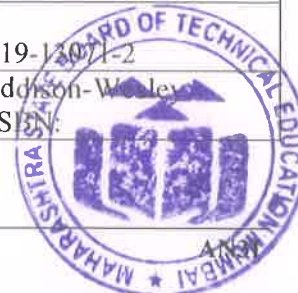
The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. Each student will have to maintain dated work diary consisting of individual 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) Develop an application to create tic-tac-toe game.
- c) Develop library management system which will able to:
  - i) Add
  - ii) Delete
  - iii) Update
  - iv) Display books related information like book no, book name, date of issue, date of deposited, and student details etc.
- d) Develop Python application for performing following operations
  - i. Deposit
  - ii. Withdraw
  - iii. Balance enquiry.
 Select appropriate data structure.
- e) Develop Python application that creates tree to store given data set using Link list representation. Locate and display specific data from data set.
- f) Any other micro-projects suggested by subject faculty on similar line. (Use functions, Classes, Objects and other features of 'Python' to develop applications.

## 13. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication
1	Data Structures and Algorithms in Python	Michael T. Goodrich, Roberto Tamassia Michael H. Goldwasser	Wiley India Pvt. Ltd. ISBN: 978-93-5424-786-6
2	Learning Python	Lutz, Mark	5th Edition, O'Reilly Publication ISBN-13: 978-1449355739
3	Python Programing	Rao, K. Nageswara Shaikh Akbar	Scitech Publications (India) Pvt. Ltd. ISBN: 9789385983450
4	Data Structures and Algorithms with Python	Kent D. Lee Steve Hubbard	Springer ISBN: 978-3-319-13071-2
5	Python Essential Reference	Beazley, David	4th Edition, Addison-Wesley Professional, ISBN: 978-0-13-135970-4





S. No.	Title of Book	Author	Publication
			9780672329784
6	Head First Python, 2nd Edition	Paul, Barry	O'Reilly Publication, 2 <sup>nd</sup> Edition, ISBN:1491919531

**14. SOFTWARE / LEARNING WEBSITES**

- a. <https://www.tutorialspoint.com/python/index.htm>
- b. <https://www.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://www.w3resource.com/python-exercises/>
- g. <https://www.anandology.com/python-practice-book/>
- h. <https://www.edureka.co/blog/data-structures-in-python/>
- i. <https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/>



