GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)

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

Course Title: Data Structure and Application
(Course Code: 1333203)

Diploma programme in which this course is offered	Semester in which offered
Information and Communication technology	3rd Semester

1. RATIONALE

Development of application systems and software that use underlying architecture of machines efficiently and effectively requires the ability to use and manipulate various types of Data Structures and other constructs. This being a fundamental ability which is language neutral yet requires use of a language for its implementation. This is a basic course which goes along with other programming courses to develop an integrated ability to efficient software development, hence this course is very important for computer engineers.

2. COMPETENCY

The course content should be taught and implemented with the aim to develop various types of related skills leading to the achievement of the following competency:

●Implement various types of Data Structures algorithms using Python.

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 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) Comprehend the basic concepts of data structures, analysis terms, OOPs concepts, and recursive functions.
b) Apply basic operations on stack, queue and linked list data structures.
c) Apply basic operations on linked list data structure.
d) Apply different sorting and searching algorithms to the small data sets.
e) Illustrate algorithms to insert, delete and searching a node in tree.

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	-	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

Following practical outcomes (PrOs) are the sub-components of the Course Outcomes (Cos). Some of the **PrOs** marked **'*'** are compulsory, as they are crucial for that particular CO at the 'Precision Level' of Dave's Taxonomy related to 'Psychomotor Domain'.

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. required
1.	Design an student class for reading and displaying the student information, the getInfo() and displayInfo() methods will be used respectively. Where getInfo() will be a private method.	I	02
2.	Design a class Complex for adding the two complex numbers and show the use of constructor.	I	02
3.	Write Python class to represent a bank account. The class contains account holder name, account number, type of account and balance amount as the data members. The member functions are as follows: to initialize the data members, to deposit an amount, to withdraw an amount after checking balance, and to display name and balance.	I	02
4.	Write Python class to represent a Cricketer. The class contains name of cricketer, team name and run as the data members. The member functions are as follows: to initialize the data members, to set run and display run.	I	04
4.	Implement push and pop algorithms of stack using list.	II	02
5.	Implement a program to convert infix notation to postfix notation using stack.	II	02
6.	Implement a program to implement queue using list that performs following operations: enqueue, dequeue, display	II	04

7.	Implement program to perform following operation on singly linked list: a. Insert a node at the beginning of a singly linked list. b. Insert a node at the end of a singly linked list. c. Insert a node after the given node of a singly linked list. d. Insert a node before the given node of singly linked list. e. Delete a node from the beginning of a singly linked list. f. Delete a node from the end of a singly linked list. g. Count the number of nodes of a singly linked list. h. Display content of singly linked list	III	08
8.	Implement program to create and display circular linked list.	III	04
9.	Implement program to perform following operation on doubly linked list : traversal, searching, adding and removing node.	III	04
10.	Implement a python program to search a particular element from list using Linear and Binary Search.	IV	04
11.	Implement Bubble sort algorithm.	IV	02
12.	Implement Selection sort and Insertion sort algorithm.	IV	02
13.	Implement Merge sort algorithm.	IV	02
14.	Implement Quick sort algorithm.	IV	02
16.	Implement construction of binary search trees.	V	02
17.	Write a menu driven program to perform following operation on Binary Search Tree: a. Create a BST. b. Insert an element in BST. c. Pre-order traversal of BST. d. In-order traversal of BST. e. Post-order traversal of BST. f. Delete an element from BST	V	08
			56 Hrs.

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. Care must be taken in assigning and assessing study report as it is a first year study report. Study report, data collection and analysis report must be assigned in a group. Teacher has to discuss about type of data (which and why) before group start their market survey.

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	Correctness of program & approach to implement logic	30
2	Readability and documentation of the program/Quality of input and output displayed (messaging and formatting)	10
3	Use python concepts to implement efficient program	20
4	Debugging ability	20
5	Program execution/answer to sample questions	20
Total		**100**

6. MAJOR EQUIPMENTS/ INSTRUMENTS REQUIRED

These major equipment's with broad specifications for the PrOs is a guide to procure them by the administrators to user in uniformity of practical's 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 or higher	All
2	Python IDEs and Code Editors Open Source : IDLE, Jupyter, Spyder (Anaconda)	All

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) Work as a leader/a team member.
b) Follow ethical practices.
c) Practice environment friendly methods and processes.
d) Follow safety precautions.

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

The major underpinning theory is given below based on the higher level UOs of *Revised Bloom's taxonomy* that are formulated for development of the COs and competency. If required, more such UOs could be included by the course teacher to focus on attainment of COs and competency.

Unit	Unit Outcomes (UOs) (4 to 6 UOs at different levels)	Topics and Sub-topics
Unit – I **Basic Concepts of Data Structures & OOP**	1a. Define linear and non-linear data structures. 1b. Define time complexity and space complexity. 1c. Explain concepts of Object Oriented programming. 1d. Explain the concept of class and object. 1e. Explain the concept of constructors and how they are used to initialize objects. 1f. Describe the Operations on arrays. 1g. Differentiate Array and List	1.1 Data Structure Basic Concepts 1.2 Types of data structures 1.3 Analysis Terms (for the definitions purpose only) : Time Complexity Space Complexity Asymptotic Notations, Big 'O', Notation , Best case Time Complexity, Average case Time Complexity, Worst case Time Complexity 1.4 OOPs Concepts 1.5 Class and Object 1.6 Constructors 1.7 Types of methods: Instance method, Class method, static method 1.8 Overview of Abstract data types 1.9 Concept of Recursive functions
Unit– II Stack and Queues	2a. Implement Stack Operations using List. 2b. List applications of Stack. 2c. Convert the given expression from Infix to Prefix/Postfix using stack. 2d. Evaluate the postfix expression using stack. 2e. Implement Queue Operations using List. 2f. Explain concepts of Circular queue. 2g. List applications of Queue. 2h. Differentiate circular and simple queues.	2.1 Overview of Stack 2.2 Operations on Stack - Push, Pop 2.3 Implementation of Stack using List 2.4 Application of Stack - Infix, Prefix and Postfix Forms of Expressions, Evaluations of postfix expression, Recursive Functions (Factorial, Fibonacci series) 2.5 Overview of Queue 2.6 Operations on Queue - Enqueue and Dequeue 2.7 Implementation of Queue using List 2.8 Limitation of Single Queue 2.9 Concepts of Circular Queue 2.10 Application of queue 2.11 Differentiate circular queue and simple queue

Unit– III **Linked List**	3a. Define a linked list. 3b. List types of Linked List. 3c. Implement basic operations on singly linked lists. 3d. Explain concepts of circular linked lists. 3e. Differentiate between circular linked list and singly linked list. 3f. Explain concepts of doubly linked lists. 3g. List applications of Linked List.	3.1 Overview of Linked list 3.2 Types of Linked List 3.3 Basic operations on singly linked list: Insertion of a new node in the beginning of the list, at the end of the list, after a given node, before a given node, Deleting the first and last node from a linked list, Count the number of nodes in linked list. 3.4 Overview of circular linked list 3.5 Difference between circular linked list and singly linked list 3.6 Overview of doubly linked list 3.7 Applications of linked list
Unit– IV **Searching and Sorting**	4a. Design and Implement search algorithms. 4b. Arrange data in ascending and descending orders using appropriate sorting algorithms. 4c. Explain the working of the given sorting method step-by-step with An example and small data set.	4.1 Searching an element into List: Linear Search, Binary Search 4.2 Sorting Methods: Bubble Sort, Selection Sort, Quick Sort, Insertion Sort, Merge Sort
Unit– V **Trees**	5a. Describe a binary tree. 5b. Draw binary search tree for the given data set. 5c. Write algorithms to traverse the tree using the given method. 5d. List applications of trees.	5.1 Binary trees: Complete Binary Tree, Basic Terms: level number, degree, in-degree and out-degree, leaf node 5.2 Binary Search Tree: Insertion of a node in binary tree, Deletion of a node in binary tree, Searching a node in binary tree. 5.3 Tree Traversal: Inorder, Preorder, Postorder 5.4 Applications of Binary tree

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 Concepts of Data Structures & OOP	8	4	4	4	12
II	Stack and Queues	8	2	6	4	12
III	Linked List	10	4	8	6	18
IV	Searching and Sorting	8	2	7	4	13
V	Trees	8	2	7	6	15
Total		**42**	**14**	**32**	**24**	**70**

Legends: *R=Remember, U=Understand, A=Apply and above (Revised Bloom's taxonomy)*

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 small reports (of 1 to 5 pages for each activity). For micro project report should be as per suggested format, for other activities students and teachers together can decide the format of the report. Students should also collect/record physical evidences such as photographs/videos of the activities for their (student's) portfolio which will be useful for their placement interviews:

a) Provide students with code snippets and ask them to identify the Big O notation for each one.

b) Undertake micro-projects in teams.

c) https://code.org/, an hour of code may be organized and students are encouraged to participate

d) Students are encouraged to register themselves in various MOOCs such as: Swayam, edx, Coursera, Udemy etc. to further enhance their learning.

e) List the applications which are developed using Python data structures

f) Encourage students to participate in different coding competitions like hackathon, online competitions on code chef etc.

g) Encourage students to form a coding club at institute level and can help the slow learners.

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) in undertaking micro-projects.
c) Managing Learning Environment
d) Diagnosing Essential Missed Learning concepts that will help for students.
e) Guide Students to do Personalized learning so that students can understand the course material at his or her pace.
f) Encourage students to do Group learning by sharing so that teaching can easily be enhanced.
g) ***'L" in section No. 4*** means different types of teaching methods that are to be employed by teachers to develop the outcomes.
h) 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.
i) With respect to ***section No.10***, teachers need to ensure to create opportunities and provisions for ***co-curricular activities***.
j) Guide students on how to address issues on environment and sustainability using the knowledge of this course

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 (group of 3 to 5). However, **in the fifth and sixth semesters**, 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 work load on each student due to the micro-project should be about ***16 (sixteen) student engagement hours*** (i.e., about one hour per week) during the course. The students ought to submit micro-project by the end of the semester (so that they develop the industry-oriented COs).

A suggestive list of micro-projects is given here. This should relate highly with competency of the course and the COs. Similar micro-projects could be added by the concerned course teacher:

a) Develop a Python Program that evaluate the given arithmetic expression using stack.
b) Develop a Python Program that maintain a queue of persons. In this queue user can add a person, delete a person and search a person.

c) Develop a Python Program that perform banking operations like withdraw cash, deposit cash and mini statement using appropriate data structure.
d) Develop a Python Program for process management algorithm by using appropriate data structure.
e) Develop a Python Program for print spooler using appropriate Data structure.
f) Develop a Python Program for Telephone Directory system. In this user can adding, searching, modifying, listing, and deleting records through the use of appropriate data structure.

g) **Phone directory application using doubly-linked lists-** This project can demonstrate the working of contact book applications and also teach you about data structures like arrays, linked lists, stacks, and queues. Typically, phone book management encompasses searching, sorting, and deleting operations. A distinctive feature of the search queries here is that the user sees suggestions from the contact list after entering each character.
h) **Hangman Game:** The Hangman program randomly selects a secret word from a list of secret words. A random word (E.g. a fruit name) is picked up from our collection and the player gets limited chances to win the game. When a letter in that word is guessed correctly, that letter position in the word is made visible. In this way, all letters of the word are to be guessed before all the chances are over.
i) **Stack and queue implementation using linked list:** Develop a python program that implements stack and queue operations using linked list representation.

13. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication with place, year and ISBN
1	Data structures and algorithms in python	Goodrich	Wiley, 2013 ISBN: 978-1-118-29027-9
2	Introduction to Programming using Python	Y Daniel Liang	Pearson, 2012, ISBN 978-0132747189
3	An Introduction to Data Structures with Applications	Jean-Paul Tremblay & Paul G. Sorenson	Tata McGraw Hill,2017, ISBN 978-0074624715
4	Python Data Structures and Algorithms	Benjamin Baka, David Julian	Packt Publishing, 2017, ISBN 978-1-78646-735-5
5	Data Structures and Algorithms using Python	Rance D. Necaise	Wiley, 2011, ISBN 978-0-470-61829-5
6	Python 3 Object oriented programming	Dusty Phillips	Packt Publishing, 2015, ISBN 978-1-78439-878-1

14. SOFTWARE/LEARNING WEBSITES

1. https://docs.python.org/3/tutorial/datastructures.html
2. http://interactivepython.org/runestone/static/pythonds/index.html

3. http://www.tutorialspoint.com/data_structures_algorithms
4. http://www.geeksforgeeks.org/data-structures/
5. http://www.studytonight.com/data-structures/
6. http://www.coursera.org/specializations/data-structures-algorithms
7. https://nptel.ac.in/courses/106106133 (Programming, Data structures and Algorithms, IIT Madras)
8. https://www.codecademy.com/learn/linear-data-structures
9. https://www.udacity.com/course/data-structures-and-algorithms-in-python--ud513
10. https://www.edx.org/learn/data-structures
11. https://www.sololearn.com/learning/1159

15. PO-COMPETENCY-CO MAPPING

Semester III	Data Structures & Applications						
	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 & Testing	PO 5 Engineering practices for society, sustainability & environment	PO 6 Project Management	PO 7 Life-long learning
Competency	**Implement various types of data structures algorithms using python**						
Course Outcomes CO a) Comprehend the basic concepts of data structures, analysis terms, OOPs concepts, and recursive functions.	3	1	1	2	-	1	1
CO b)Apply basic operations on stack, queue data structures.	3	2	2	3	-	2	1
CO c)Apply basic operations on linked list data structures.	3	2	2	3	-	2	1
CO d)Apply different sorting and searching algorithms to the small data sets.	3	2	2	3	-	2	1
CO e)Illustrate algorithms to insert,	3	2	2	3	-	2	1

delete and searching a node in tree.							

*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

S. No.	Name and Designation	Institute	Contact No.	Email
1	Shri P. P. Kotak Principal	S & SS Gandhi College Surat	8200601748	kotakp2003@yahoo.com
2	Smt. M. P. Mehta Head of the Department	Government Polytechnic, Gandhinagar	987958273	manishamehtain@gmail.com
3	Ms. Darshita S. Pathak	A.V.P.T.I. Rajkot	9879251273	dsp.pathak@gmail.com
4	Mr. Ashish Patel	Government Polytechnic for Girls, Surat	9924050239	gpgsecamp@gm ail.com