

Programme Name/s : Artificial Intelligence/ Artificial Intelligence and Machine Learning/ Data Sciences
Programme Code : AI/ AN/ DS
Semester : Third
Course Title : DATA STRUCTURE USING PYTHON
Course Code : 313306

I. RATIONALE

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

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME

The aim of this course is to help the student to attain the following Industry Identified Outcomes through various teaching learning experiences:

- Implement Data Structures using Python to solve problems.

III. COURSE LEVEL LEARNING OUTCOMES (COS)

Students will be able to achieve & demonstrate the following COs on completion of course based learning

- CO1 - Develop Python program using basic syntactical constructs.
- CO2 - Perform operations on sequence structures in Python.
- CO3 - Implement Modules, Packages in Python for given problem.
- CO4 - Design classes for given problem.
- CO5 - Implement Linear Data Structure in Python.
- CO6 - Develop Python program to implement tree data structure.

IV. TEACHING-LEARNING & ASSESSMENT SCHEME

Course Code	Course Title	Abbr	Course Category/s	Learning Scheme					Credits	Paper Duration	Assessment Scheme										Total Marks
				Actual Contact Hrs./Week			SL	LH			NLH	Theory			Based on LL & TL		Based on SL				
				CL	TL	LL						FA-TH	SA-TH	Total	Practical		SLA				
				Max	Max	Max	Min	Max			Min				Max	Min	Max	Min			
313306	DATA STRUCTURE USING PYTHON	DSP	DSC	3	-	4	1	8	4	3	30	70	100	40	25	10	25@	10	25	10	175

Total IKS Hrs for Sem. : 0 Hrs

Abbreviations: CL- Classroom Learning , TL- Tutorial Learning, LL-Laboratory Learning, SLH-Self Learning Hours, NLH-Notional Learning Hours, FA - Formative Assessment, SA -Summative assessment, IKS - Indian Knowledge System, SLA - Self Learning Assessment

Legends: @ Internal Assessment, # External Assessment, *# On Line Examination , @\$ Internal Online Examination

Note :

1. FA-TH represents average of two class tests of 30 marks each conducted during the semester.
2. If candidate is not securing minimum passing marks in FA-PR of any course then the candidate shall be declared as "Detained" in that semester.
3. If candidate is not securing minimum passing marks in SLA of any course then the candidate shall be declared as fail and will have to repeat and resubmit SLA work.
4. Notional Learning hours for the semester are (CL+LL+TL+SL)hrs.* 15 Weeks
5. 1 credit is equivalent to 30 Notional hrs.
6. * Self learning hours shall not be reflected in the Time Table.
7. * Self learning includes micro project / assignment / other activities.

V. THEORY LEARNING OUTCOMES AND ALIGNED COURSE CONTENT

Sr.No	Theory Learning Outcomes (TLO's)aligned to CO's.	Learning content mapped with Theory Learning Outcomes (TLO's) and CO's.	Suggested Learning Pedagogies.
1	TLO 1.1 Describe the given variables, keywords and constants in Python. TLO 1.2 Use indentation, comments in the given program. TLO 1.3 Use different types of operators for writing arithmetic expressions. TLO 1.4 Write Python programs using control flow.	Unit - I Introduction and Control Flow statements in Python 1.1 Features of Python - interactive, object oriented, interpreted, platform independent. 1.2 Python building blocks - identifiers, keywords, indentation, 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.	Presentations Chalk-Board Hands-on

Sr.No	Theory Learning Outcomes (TLO's) aligned to CO's.	Learning content mapped with Theory Learning Outcomes (TLO's) and CO's.	Suggested Learning Pedagogies.
2	<p>TLO 2.1 Develop Python program to manipulate lists for the given problem.</p> <p>TLO 2.2 Develop Python program to manipulate tuples for the given problem.</p> <p>TLO 2.3 Write Python program to manipulate sets for the given problem.</p> <p>TLO 2.4 Write Python program to manipulate dictionaries for the given problem.</p> <p>TLO 2.5 Develop relevant user defined functions for the given problem.</p>	<p>Unit - II Python specific Data Structures and functions</p> <p>2.1 Lists: Defining lists, accessing values in list, deleting values in list, updating lists, basic list operations, built - in list functions.</p> <p>2.2 Tuples: Accessing values in tuples, deleting values in tuples, and updating tuples, basic tuple operations, built - in tuple functions.</p> <p>2.3 Sets: Accessing values in set, deleting values in set and updating sets, basic set operations, built - in set functions.</p> <p>2.4 Dictionaries: Accessing values in dictionary, deleting values in dictionary and updating dictionary, basic dictionary operations, built- in dictionaries functions.</p> <p>2.5 Use of Python built- in functions (e.g.type/ data conversion functions, math functions etc.)</p> <p>2.6 User defined functions: Function definition, function calling, function arguments and parameter passing, return statement, scope of variables: global variable and local variable.</p>	Presentations Chalk-Board Hands-on
3	<p>TLO 3.1 Create Python module for the given problem.</p> <p>TLO 3.2 Develop Python package for the given problem.</p> <p>TLO 3.3 Use NumPy for performing mathematical operations on arrays.</p>	<p>Unit - III Python Modules and Packages</p> <p>3.1 Modules: writing modules, importing modules, importing objects from modules, Python built- in modules (e.g. numeric and mathematical module, functional programming module).</p> <p>3.2 Python packages: introduction, writing Python packages.</p> <p>3.3 Using standard Numpy: methods in Numpy, creating arrays and initializing, reading arrays from files, special initializing functions, slicing and indexing, reshaping arrays, combining arrays, NumPy maths.</p>	Chalk-Board Presentations Hands-on
4	<p>TLO 4.1 Apply the basic concepts of object oriented programming to define classes and objects.</p> <p>TLO 4.2 Implement the concept of polymorphism</p> <p>TLO 4.3 Create Python program using inheritance for the given problem</p>	<p>Unit - IV Object Oriented Programming in Python</p> <p>4.1 Introduction to object oriented programming, creating classes and objects, constructor and destructor in Python.</p> <p>4.2 Data abstraction and data encapsulation.</p> <p>4.3 Concept of polymorphism- method overloading and overriding.</p> <p>4.4 Inheritance and types of inheritance.</p>	Chalk-Board Presentations Hands-on

Sr.No	Theory Learning Outcomes (TLO's) aligned to CO's.	Learning content mapped with Theory Learning Outcomes (TLO's) and CO's.	Suggested Learning Pedagogies.
5	TLO 5.1 Describe linear data structures in Python. TLO 5.2 Develop Python code using arrays for given problem. TLO 5.3 Write Python code to implement link list for given problem. TLO 5.4 Implement stack in Python program. TLO 5.5 Implement queues in Python program.	Unit - V Linear Data Structure Arrays, Link List, Stack and Queues using Python. 5.1 Data Structures – definition, linear data structures, non-linear data structures arrays - overview, types of arrays, operations on arrays, arrays vs list. 5.2 Searching -linear search and binary search, sorting - bubble sort, insertion sort. 5.3 Linked Lists – singly linked list, doubly linked list, circular linked lists, implementation using Python packages for link list. 5.4 Stacks : introduction to stacks, stack applications - expression evaluation, backtracking, traversal - infix, prefix and postfix concepts. 5.5 Queues: introduction to queues, queue applications - breadth first search, depth first search.	Presentations Chalk-Board Flipped Classroom
6	TLO 6.1 Implement various types of trees using Python code. TLO 6.2 Perform various operations on trees using Python code.	Unit - VI Non-Linear Data Structure. 6.1 Trees - tree Terminology, binary trees: implementation, tree traversals, binary search trees 6.2 Applications of trees - spanning tree, BST , tree traversal - inorder, preorder and postorder concepts.	Presentations Hands-on

VI. LABORATORY LEARNING OUTCOME AND ALIGNED PRACTICAL / TUTORIAL EXPERIENCES.

Practical / Tutorial / Laboratory Learning Outcome (LLO)	Sr No	Laboratory Experiment / Practical Titles / Tutorial Titles	Number of hrs.	Relevant COs
LLO 1.1 Install Python integrated development environment.	1	* a) Install and configure Python IDE . b) Write Python program to display message on screen	2	CO1
LLO 2.1 Implement basic operators in Python.	2	* Write simple Python program using operators: a) Arithmetic operators b) Logical operators c) Relational operators d) Conditional operators e) Bitwise operators f) Ternary operator	2	CO1
LLO 3.1 Implement control flow operations for solving given task.	3	* Write simple Python program to demonstrate use of conditional statements : i) if ii) if...else iii) Nested if iv) Switch case	4	CO1
LLO 4.1 Implement while control loop for solving iterative problems.	4	Develop a simple Python program to demonstrate use of control loop: while	2	CO1

Practical / Tutorial / Laboratory Learning Outcome (LLO)	Sr No	Laboratory Experiment / Practical Titles / Tutorial Titles	Number of hrs.	Relevant COs
LLO 5.1 Implement for loop for solving iterative problems.	5	* Create a simple program to demonstrate use of for loop in Python (e.g : various pattern building, printing multiplication table, checking palindrome number etc.)	2	CO1
LLO 6.1 Perform basic operations on the lists.	6	* Write Python program to perform following operations on lists : a) Create b) Access c) Update d) Delete elements in list	2	CO2
LLO 7.1 Execute various tuple operations.	7	Develop Python program to perform following operations on tuples: a) Create b) Access c) Update d) Delete tuple elements	2	CO2
LLO 8.1 Implement various set operations.	8	* Write Python program to perform following operations on set : a) Create b) Access c) Update d) Delete access set elements	2	CO2
LLO 9.1 Execute various operations on dictionaries.	9	* Create a program to perform following operations on dictionaries in Python: a) Create b) Access c) Update d) Delete e) Looping through dictionary	2	CO2
LLO 10.1 Use built-in math functions in Python.	10	Apply math built - in function in Python.	2	CO2
LLO 11.1 Create user-defined functions to solve the given task	11	* Develop user defined Python function for given problem: write a function with minimum 2 arguments	2	CO2
LLO 12.1 Apply built-in mathematical modules from python.	12	Create a program to demonstrate use of built-in module (e.g. numeric, mathematical functional and programming module) in Python.	2	CO3
LLO 13.1 Write user-defined module in Python.	13	* Write program to ceate a user - defined module (e.g : building calculator) in Python.	2	CO3
LLO 14.1 Apply built-in packages from Python.	14	* Develop Python program to demonstrate use of NumPy packages for creating , accessing and performing different array operations.	2	CO4
LLO 15.1 Create user-defined packages in Python.	15	Write a program to create user defined packages in Python.	2	CO4
LLO 16.1 Implement the concept of polymorphism in Python.	16	Write program in Python to demonstrate following operations: a) Method overloading b) Method overriding	2	CO4

Practical / Tutorial / Laboratory Learning Outcome (LLO)	Sr No	Laboratory Experiment / Practical Titles / Tutorial Titles	Number of hrs.	Relevant COs
LLO 17.1 Implement the concept of inheritance in Python.	17	* Develop program in Python to demonstrate following operations: a) Single inheritance b) Multilevel inheritance c) Multiple inheritance d) Hybrid inheritance e) Hierarchical inheritance	4	CO4
LLO 18.1 Execute the array operations using Python.	18	* 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	4	CO5
LLO 19.1 Implement the linked list operations using Python.	19	* Write a program in Python for linked list to demonstrate following operations : a) Insertion b) Deletion c) Updating d) Merging to list	4	CO5
LLO 20.1 Implement the queue operations using Python.	20	* Write a program in Python to demonstrate queue data structure operations : a) Enqueue b) Dequeue c) Display	4	CO5
LLO 21.1 Implement searching techniques in Python.	21	* Use the searching techniques in data structures: a) Linear Search b) Binary Search	2	CO5
LLO 22.1 Implement sorting techniques in Python.	22	* Write a Python program to implement following sorting techniques: a) Bubble Sort b) Insertion Sort	4	CO5
LLO 23.1 Implement stack data structure in Python.	23	* Write a program in Python to evaluate an expression.	2	CO6
LLO 24.1 Implement sorting techniques in Python.	24	Write a program to create binary tree from the given list using binary tree module in Python.	2	CO6
Note : Out of above suggestive LLOs - <ul style="list-style-type: none"> '*' Marked Practicals (LLOs) Are mandatory. Minimum 80% of above list of lab experiment are to be performed. Judicial mix of LLOs are to be performed to achieve desired outcomes. 				

VII. SUGGESTED MICRO PROJECT / ASSIGNMENT/ ACTIVITIES FOR SPECIFIC LEARNING / SKILLS DEVELOPMENT (SELF LEARNING)

Micro project

- The micro project has to be industry based, internet based ,workshop based, laboratory based or field based as suggested by teacher.
- a) Develop an application to create tic-tac-toe game.
- b) Create a code generator. This can that take text as input, replaces each letter with another letter, and outputs the “encoded” message.
- c) Build an interactive quiz. Build a personality or recommendation quiz that asks users some questions, stores their answers, and then performs some kind of calculation to give the user a personalized result based on their answers.
- d) Build an alarm clock using snooze function.

Other

- Implement various applications of stacks.
- Compute the arithmetic mean along the specified axis using NumPy.
- Present a seminar on different IDE used in Python.

Note :

- Above is just a suggestive list of microprojects and assignments; faculty must prepare their own bank of microprojects, assignments, and activities in a similar way.
- The faculty must allocate judicious mix of tasks, considering the weaknesses and / strengths of the student in acquiring the desired skills.
- If a microproject is assigned, it is expected to be completed as a group activity.
- SLA marks shall be awarded as per the continuous assessment record.
- If the course does not have associated SLA component, above suggestive listings is applicable to Tutorials and maybe considered for FA-PR evaluations.

VIII. LABORATORY EQUIPMENT / INSTRUMENTS / TOOLS / SOFTWARE REQUIRED

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Hardware : Personal Computer , RAM minimum 2 GB onwards. Operating System : Windows 10 onwards Software : Any relevant Python IDE like IDLE/PyCharm/Spyder/PyDev etc.	All

IX. SUGGESTED WEIGHTAGE TO LEARNING EFFORTS & ASSESSMENT PURPOSE (Specification Table)

Sr.No	Unit	Unit Title	Aligned COs	Learning Hours	R-Level	U-Level	A-Level	Total Marks
1	I	Introduction and Control Flow statements in Python	CO1	6	2	4	6	12
2	II	Python specific Data Structures and functions	CO2	7	2	4	6	12
3	III	Python Modules and Packages	CO3	6	2	4	6	12
4	IV	Object Oriented Programming in Python	CO4	6	2	4	4	10
5	V	Linear Data Structure Arrays, Link List, Stack and Queues using Python.	CO5	12	2	4	8	14
6	VI	Non-Linear Data Structure.	CO6	8	2	4	4	10
Grand Total				45	12	24	34	70

X. ASSESSMENT METHODOLOGIES/TOOLS

Formative assessment (Assessment for Learning)

- Continuous Assessment based on Process and Product related Performance Indicators. Each Practical will be assessed considering:
60% weightage is to Process.
40% weightage is to Product.

Summative Assessment (Assessment of Learning)

- Laboratory Performance, Viva Voce

XI. SUGGESTED COS - POS MATRIX FORM

Course Outcomes (COs)	Programme Outcomes (POs)							Programme Specific Outcomes* (PSOs)		
	PO-1 Basic and Discipline Specific Knowledge	PO-2 Problem Analysis	PO-3 Design/ Development of Solutions	PO-4 Engineering Tools	PO-5 Engineering Practices for Society, Sustainability and Environment	PO-6 Project Management	PO-7 Life Long Learning	PSO-1	PSO-2	PSO-3
CO1	3	-	-	1	-	-	-			
CO2	2	2	2	1	-	-	1			
CO3	-	2	2	1	-	-	1			
CO4	2	2	2	1	1	2	1			
CO5	2	2	2	1	1	2	1			
CO6	2	2	2	1	-	2	1			

Legends :- High:03, Medium:02,Low:01, No Mapping: -
*PSOs are to be formulated at institute level

XII. SUGGESTED LEARNING MATERIALS / BOOKS

Sr.No	Author	Title	Publisher with ISBN Number
1	Rao, K. Nageswara Shaikh Akbar	Python Programing	Scitech Publications (India) Pvt. Ltd. ISBN: 9789385983450
2	Michael T. Goodrich, Roberto Tamassia	Data Structures and Algorithms in Python	Wiley publication New Delhi 2016 ISBN: 978-81- 265-5423-2
3	Beazley, David	Python Essential Reference	4th Edition, Addison-Wesley Professional, ISBN: 9780672329784
4	Dr.Basant Agarwal	Hands-On Data Structures and algorithms with Python	3rd Edition, Packt publisher, ISBN : 9781801073448

XIII . LEARNING WEBSITES & PORTALS

Sr.No	Link / Portal	Description
1	https://onlinecourses.nptel.ac.in/noc22_cs26/preview	Online Learning Initiatives by IIT faculties.
2	https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/	Data structures in Python

Sr.No	Link / Portal	Description
3	https://www.freecodecamp.org/news/object-oriented-programming-in-python/	Object Oriented Programming Concepts
4	https://realpython.com/python3-object-oriented-programming/	Basics Object Oriented Programming Concepts.
Note : <ul style="list-style-type: none">Teachers are requested to check the creative common license status/financial implications of the suggested online educational resources before use by the students		