MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(Deemed to be University)
Affiliated to JNTUA, Ananthapuramu & Approved by AICTE, New Delhi
NAAC Accredited with A+ Grade, NIRF India Rankings 2025 - Band: 201-300 (Engg.)
NBA Accredited - B.Tech. (CIVIL, CSE, ECE, EEE, MECH, CST), MBA & MCA

A Report on Twenty Days External Technical Training on "Python and Java with Data Structures" Organized by Department of Computer Applications in association with M/s. Tap Academy, Bengaluru from 27.04.26 to 01.05.26

Coordinator: Dr. M.Saravanamuthu, Associate Professor, Department of Computer Applications
Duration: From 9.30am to 4.30pm
Mode of conduct: Offline
Venues: KKB101 for Python with 66 students and KKB108 for Java with 63 students in parallel sessions
Target Audience: I-MCA students

Resource Persons:
1. Mr. Mallikarjuna for Python
2. Mr. Vamsi and Mr.Manjunath for Java
Total Participants: 145 (students-129 and staff-16)

About the training:
On day-1 the technical training has been initiated with two batches, python and java at the respective venues. Every day the training had four sessions from 9.30am to 11.00am, 11.10 to 01.10pm, 2.00pm to 3.20pm and 3.30pm to 4.30pm. Both the trainers explained the concepts through augmented reality in a visual model which makes easier for students, understanding the harder concepts in a simple way. Every day the first three sessions are used for concept explanation and the last session is utilized for hands-on practice and making students to do coding themselves.

Every day assignments are given to the students on the topics discussed as a home work through Tap Academy's portal. The assessments have been given on alternate days from second of week onwards with proctored monitoring as in actual drives. Each session has been monitored by department faculty with a separate schedule.
The first 16 days are covered with the fundamentals of Python / Java as per the topics given below,

Day	Python Topics	Java Topics
1.	Interview Process Overview What is an Interview Interview Roadmap Introduction to Programming What is programming? Types of programming languages	Interview Process Overview What is an Interview Interview Roadmap Introduction to Programming What is programming? Types of programming languages
2.	History and Features of Programming Language Exploring various features Platform independence • Portability and benefits Python program structure Variables and syntax Indentation and program flow	History and Features of Programming Language Exploring various Features Platform independence Portability and benefits Fundamentals and Structure Main method, class, object Syntax and program flow

	Object-Oriented Overview	Object-Oriented Overview
3.	Modular Programming Pattern printing using loops Python data types (int, float, string, list, tuple) Arithmetic operators	Modular Programming & Pattern Using loop for printing patterns Data Types and Operators Primitive data types (int, char, etc.) Arithmetic operations
4.	Type conversion in Python Variable scope (local & global) Unicode basics Conditional statements (if, if-else, nested if) Ternary operator Grade mapping example	Type Casting & Variable Scope Widening vs. narrowing conversion Variable scope in methods and classes Unicode & ASCII basics Conditional Statments If, if-else, nested ifs Ternary operator Grade mapping example
5.	Functions in Python Code reusability for loop while loop Prime number program Multiplication tables	Methods and Code Reusability Creating and using methods Benefits of reusing code Loops For, while, do-while loops Prime number check, multiplication tables
6.	Count digits Sum of digits Reverse numbers Palindrome check HCF and LCM Common multiples Euclid's algorithm	Digit & Number Analysis Count digits, sum of digits Reverse numbers, check palindrome Mathematical Computations HCF & LCM Common multiples Euclid's algorithm
7.	Lists and nested lists Min, max, sum Second largest element Pair processing Frequency analysis Sorting Unique and repeated elements	Working with Arrays 1D, 2D, 3D Min Max sum Second Largest Elements Array Operations Pair processing Frequency analysis Sorting, unique/repeated elements
8.	Merge lists Frequency of elements Subarrays / sublists Longest consecutive sequence Rearranging lists	Advanced Array Concepts Merge two arrays Frequency of elements Subarrays and Rearrange array Fixed-length and variable-length subarrays Longest consecutive subarray
9.	Creating strings in Python String immutability Comparing strings Substring generation Palindromic substrings String comparison	String Introduction Various ways to create strings in Java Understanding how strings are stored in memory Different methods to compare strings Advanced String Operations Substring generation Find palindromic substrings Comparing strings
10.	Splitting strings Reversing words Real-time string examples Default arguments (function overloading concept) Encapsulation Access control	String Processing Splitting strings Reversing word order Real-time string array examples Method Overloading and Encapsulation Overloading methods Encapsulation and access control
11.	Constructors using __init__() Object initialization Public, protected, private concepts Class variables vs instance variables	Constructors and Local Chaining Reuse constructors using this() Practical examples Static and Access Modifiers static keyword usage public, private, protected modifiers
12.	Inheritance in Python Using super() Method overriding Runtime polymorphism	Inheritance and super() Types of inheritance Using super for constructor and method access Constructor Chaining and Polymorphism this() and super() usage Constructor overloading Compile-time and runtime polymorphism Method overriding

13.	Aggregation and composition Code reuse Abstract classes (ABC module) Interface-like implementation	Aggregation and Composition Has-a relationships Code reuse through composition Abstraction and Interfaces Abstract classes Interface-based abstraction Creating interfaces Interface vs Abstract class
14.	Exception handling importance Handling errors in Python try-except-finally Multiple exceptions Built-in exceptions	Exception Handling Introduction Importance of exception handling Different ways of handling Exception Exception Hierarchy try-catch-finally Multiple catch blocks Checked vs Unchecked exceptions
15.	Introduction to design patterns OOP design examples List, Tuple, Set, Dictionary Stack and Queue implementation	Design Pattern Introduction Introduction to OOP design patterns Examples and use cases Collections: ArrayList, LinkedList and Stack, Queue Introduction to List interface Implementing stack and queue Use cases and methods
16.	Set operations	Set and Map Programs using Set (duplicates removal, set operations) Key-value mapping HashMap, TreeMap use cases

The last four days are covered with Data Structures using Python and Java. The topics are shown below,

Day	Data Structures using Python and Java
17.	Binary search Span of an array ceil and floor of a sorted array wood cutting Program count of smaller and equal elements Painters partition
18.	Stack Data Structure Introduction Stack implementation using Linked List peek() isEmpty() size() Time Complexity Balanced Parenthesis Queue Data Structure Introduction Applications of Queue Creating Queue Enqueue Dequeue getFront(), getRear() isEmpty(), isFull()
19.	Linked List Introduction Creating a Linked List Properties of Linked List Adding An Element to Linked List Add First

	Add element in Linked List using index Add All
20.	Doubly Linked List Doubly Linked List_add() Doubly Linked List_printAllElements() Doubly Linked List_printReverse Doubly Linked List_Add First Doubly Linked List_Add element in Linked List using index Doubly Linked List_ Add All

On every day the trainers had meeting with training coordinator and class in-charges for a discussion about students' cooperation and for any issues on the day.

Participants feedback:

Participants have gained knowledge of foundations of python, java and data structures models.
Most of the students replied that the external training is very useful for facing forthcoming campus drives. The daily assignments helped the participants to practice their programming skills. Further the proctored assessments motivated them in coding in python and java languages. The last four days enabled them to improve the programming using data structure models.

Outcome of the workshop:

The result of the twenty days external training equipped the participants in the following areas

- Foundations of python and java programming languages
- Periodical assignments for hands-on practice
- Improvement in coding skills in python and java
- Gained confident in programming with data structures
- Students are now ready for drives after this training