COLLECTIONS IN JAVA

Event No	AD001
Organizing Department	Artificial Intelligence and Data Science
Date	25/04/2024
Time	09:00 AM to 11:00 AM
Event Type	Expert Talk
Event Level	Dept. Level
Meeting Medium	
Meeting Link	https://bit.ly/CollectionOnJava
Total Participants	78
Faculty - Internal	2
Students - Internal	76

Related SDG

Resource Persons

Sl	Type	Name	Designation	Company	Email	Phone
1	Resource Person	K Roakesh	Developer	Zoho Corporation, Chennai	roakesh.k@gmail.com	xxxxxxxxxx

Involved Staffs

Sl	Name	Role
1	Sankar Ganesh S	Coordinator
2	Sudha S V	Convenor

Outcome

The fundamental concepts behind each type of collection.How to use collections effectively to solve various programming problems.The performance implications of choosing one collection over another in different scenarios.Common operations and their time complexities for each type of collection.Iterating through collections using different methods such as iterators, enhanced for loops, or streams.How to manipulate collections using methods provided by the Java Collections Framework, such as add, remove, contains, etc.The importance of understanding generics and how they are used in collections to ensure type safety.

Event Summary

The event started with Welcome Address by Dr Sankar Ganesh S. He introduced the guest to the audience. In Java, collections are data structures used to store and manipulate groups of objects. The following classes were explained by the Resource Person.:**ArrayList**: An ordered collection that allows duplicate elements. It dynamically resizes itself when elements are added or removed.**LinkedList**: Similar to ArrayList but implemented as a doubly linked list. It provides more efficient insertion and deletion operations at the cost of slower access times.**HashSet**: An unordered collection that does not allow duplicate elements. It uses hashing to store elements, providing constant-time performance for basic operations like add, remove, and contains.**TreeSet**: A sorted set implemented using a red-black tree. It stores elements in sorted order and does not allow duplicates.**HashMap**: A hash table-based implementation of the Map interface. It stores key-value pairs and provides constant-time performance for basic operations like get and put.**TreeMap**: A sorted map implemented using a red-black tree. It stores key-value pairs in sorted order based on the natural ordering of the keys or a custom comparator.Finally the event ended with vote of thanks.

Click to View

Click to View

Click to View

***** END *****