**Core Java - Assignment**

**Module - 1**

1. **Array and Strings :**

**Theory : -**

1. One-Dimensional and Multidimensional Arrays :

-> A linear collection of elements of the same data type, stored in contiguous memory locations.

-> Elements are accessed using their index (starting from 0).

Multidimensional Array :

-> Arrays with more than one dimension, representing a table or grid of elements.

-> Often used to represent matrices or tables.

1. String Handling in Java: String Class, String-buffer, String Builder.

-> String Class :

-> Immutable: Once a String object is created, its value cannot be changed.

-> Efficient for Read-Only Operations : Well-suited for situations where you primarily need to

read and manipulate string data without modifying it directly.

-> Common Methods:

-> length() : Returns the length of the string.

-> charAt (index) : Returns the character at the specified index.

-> substring (beginlndex, endlndex) : Returns a substring of the string.

-> concat(str) : Concatenates two strings.

-> toLowerCase() , toUpperCase() : Converts the string to lowercase or uppercase.

-> equals() , equalslgnorecase() : Compares strings for equality.

-> indexOf() , lastlndexOf() : Finds the index of a character or substring.

-> split() : Splits the string into an array of substrings.

-> StringBuffer :

-> Mutable : Allows you to modify the contents of the string after it is created.

-> Slower than String for Read-Only Operations: More efficient for operations that involve

frequent modifications to the string content.

-> Common Methods:

-> append() : Appends characters or strings to the end of the buffer.

-> insert() : Inserts characters or strings at a specified position.

-> delete() : Deletes characters within a specified range.

-> replace() : Replaces characters within a specified range.

-> reverse() : Reverses the order of characters in the buffer.

-> String Builder:

-> Mutable: Similar to StringBuffer , but generally faster for single-threaded environments.

-> Not Thread-Safe: Not synchronized, making it less suitable for multi-threaded applications.

-> Common Methods: Similar to StringBuffer , with the same functionality.

Choosing the Right Class:

-> String: Use for read-only operations or when string content remains constant.

-> StringBuffer: Use for frequent modifications in multi-threaded environments.

-> StringBuilder: Use for frequent modifications in single-threaded environments for better

performance.

-> Array of Objects :

-> An array of objects In Java IS a data structure that stores a collection of objects of the same

class. Each element of the array holds a reference to an object, rather than the object itself.

-> String Methods (length, charAt, substring, etc.)

-> Java's String class offers a rich set of methods for manipulating and analyzing text. Here are

some of the most common ones:

-> length() :Returns the number of characters in the string.

-> charAt(int index) : Returns the character at the specified index.

Note: Indexing starts from O.

-> substring (int beginlndex) : Returns a substring from the specified beginlndex to the end of the string.

-> substring(int beginlndex, int endlndex) : Returns a substring from the specified beginlndex (inclusive) to endlndex (exclusive).

-> concat(String str) : Concatenates the specified string to the end of this string.

-> toLowerCase ( ) : Converts all characters in the string to lowercase.

-> toUpperCase() : Converts all characters in the string to uppercase.

-> equals (Object anObject) : Compares this string to the specified object. Returns true if they are equal; otherwise,

returns false .

-> equalsIgnoreCase(String anotherString) :Compares this string to another string, ignoring case considerations.

-> indexOf (String str) :Returns the index within this string of the first occurrence of the specified substring.

-> lastlndexOf(String str) : Returns the index within this string of the last occurrence of the specified substring.

-> split (String regex) : Splits this string around matches of the given regular expression.