NAGARJUNA COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous College under VTU)
Venkatagiri Kote post, Devanahalli, Bengaluru-562164

Department of Computer Science and Engineering

ADVANCED JAVA PROGRAMMING

LABORATORY MANUAL

21CSL55

Prepared By:
Prof. KOUSHIKA K H

Department Vision and Mission statements

VISION

Excellence in creating globally competent professionals and leaders in the field of Computer Science & Engineering.

MISSION

M1: Creating Excellence in Computer Science & Engineering education through academic professionalism, teaching, curricula which reflect the changing needs of the society.

M2: Establishing center of excellence by creating knowledge through research and industrial exposure in the area of Computer Science & Engineering.

M3: Developing communication skill, leadership qualities, teamwork & skills for continuing education among the students.

M4: Inculcating ethics, human values and skills for solving societal problems and environmental protection.

M5: Validate engineering knowledge through innovative research projects to enhance their employability and entrepreneurship skills.

ADVANCED JAVA PROGRAMMING LABORATORY

Course code	L:T:P:S	Credits	Exam Marks	Exam Duration	Course Types
21CSL55	0:0:2:0	01	CIE:50 SEE:50	03 Hours	PCC

Course Objectives:

The Student will:

- Understand and manipulate Java strings effectively.
- Master the usage of Array Lists for dynamic data storage.
- Gain proficiency in file manipulation functions.
- Understand and implement the Iterator class for efficient data traversal.
- Learn to handle byte array input efficiently in Java.

Programs List

Sl. No.	Programs
1	Write a Java program to sort array of strings using CompareTo function.
2	Write a Java program to count the occurrence of character in a given string using suitable String handling functions.
3	Write a program to perform string operations using Array List. Write functions for the following a. Append - add at end b. Insert - add at particular index c. Search d. List all string starts with given letter.
4	Write a java program to Create an ArrayList of type String and prompt that user for three names and add these names to your ArrayList. b) Print a message with the number of elements in the ArrayList for the user using the size method. c) Prompt the user for two more names and add them to the ArrayList and once again print a message with the number of elements in the ArrayList for the user. d) Use a loop to print all of the names in the ArrayList for the user. e) Ask the user for a name to remove, remove the value the user provides, and then use an enhanced for loop to print all of the names in the ArrayList for the

	user.
5	Write a Java program to add elements to the start of a list and to add elements to the end of the list. Obtain the first and last element. Remove first and last element.
6	Write a java program to implement both the Iterator and ListIterator interfaces.
7	Create a program to reads the data from two files and writes in to another file and to display only files in a specified location.
8	Write a Java program to read and write string using ByteArray Input and Output stream.

Course outcomes:

The student will be able to:

CO1: Students will master advanced string manipulation techniques in Java, including substring extraction, concatenation, formatting, and regular expressions.

CO2: Students will demonstrate the effective use of Array Lists in Java, understanding dynamic resizing, adding, removing, and iterating through elements.

CO3: Students will learn to handle reading and writing data from/to streams.

CO4: Students will learn to efficiently handle byte array input in Java, including reading and writing data from/to streams, such as File Input Stream and Byte Array Output Stream.

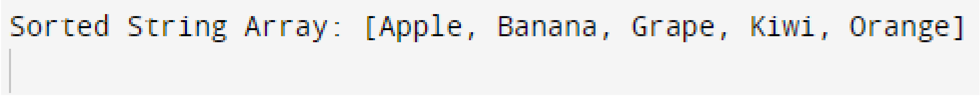
CO5: Students will understand and implement the Iterator class for traversing various data structures, including Array Lists, efficiently and effectively.

1. Write a Java program to sort array of strings using CompareTo function.

PROGRAM

```
public class StringArraySort {
public static void main(String[] args) {
// Sample array of strings
String[] stringArray = {"Banana", "Apple", "Orange", "Grape", "Kiwi"};
// Sorting the array using compareTo() function
for (int i = 0; i < stringArray.length - 1; i++) {
for (int j = i + 1; j < stringArray.length; j++) {
if (stringArray[i].compareTo(stringArray[j]) > 0) {
// Swap if the current element is greater than the next
String temp = stringArray[i];
stringArray[i] = stringArray[j];
stringArray[j] = temp;
}
}
}
// Displaying the sorted array
System.out.println("Sorted String Array: " + Arrays.toString(stringArray));
}
}
```

OUTPUT:



```
Sorted String Array: [Apple, Banana, Grape, Kiwi, Orange]
```

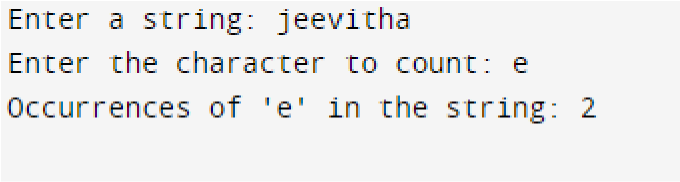
2) Write a Java program to count the occurrence of character in a given string using suitable String handling functions.

PROGRAM

```
import java.util.Scanner;
public class CharacterOccurrenceCounter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Input the string
        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();
        // Input the character to count
        System.out.print("Enter the character to count: ");
        char targetCharacter = scanner.next().charAt(0);
        // Count occurrences using string handling functions
        int count = countOccurrences(inputString, targetCharacter);
        // Display the result
        System.out.println("Occurrences of '" + targetCharacter + "' in the string: " + count);
    }
    private static int countOccurrences(String inputString, char targetCharacter) {
        int count = 0;
        for (char c : inputString.toCharArray()) {
            if (c == targetCharacter) {
                count++;
            }
        }

        return count;
    }
}
```

OUTPUT:



```
Enter a string: jeevitha
Enter the character to count: e
Occurrences of 'e' in the string: 2
```

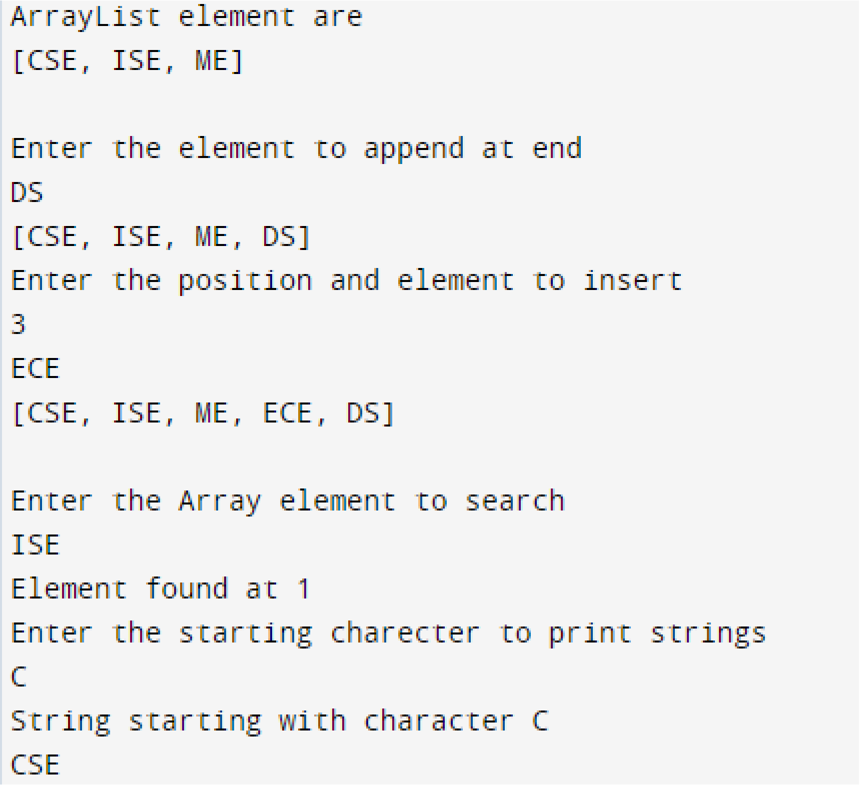
3) Write a program to perform string operations using Array List. Write functions for the following. a. Append - add at end b. Insert - add at particular index c. Search d. List all string

PROGRAM

```
import java.util.*;
public class ArrayL
{
ArrayList<String> list=new ArrayList<String>(); //Creating arraylist
public void arraydisplay()
{
list.add("CSE");//Adding object in arraylist
list.add("ISE");
list.add("ME");
System.out.println("ArrayList element are");
System.out.println(list);
System.out.println("");
}
public void appendatend()
{
System.out.println("Enter the element to append at end");
Scanner scob1=new Scanner(System.in);
String ele=scob1.next();
list.add(ele);
System.out.println(list);
System.out.println("");
}
public void insertatpos()
{
System.out.println("Enter the position and element to insert");
Scanner scob1=new Scanner(System.in);
int posind=scob1.nextInt();
String ele=scob1.next();
list.add(posind,ele);
System.out.println(list);
System.out.println("");
}
public void searchele()
{
System.out.println("Enter the Array element to search");
Scanner scobj=new Scanner(System.in);
String arele=scobj.next();
int in=list.indexOf(arele);
if(in==-1)
{
System.out.println("Element not found");
}
else
{
System.out.println("Element found at "+in);
}
}

void print()
{
System.out.println("Enter the starting charecter to print strings");
Scanner nip=new Scanner(System.in);
char inputc=nip.next().charAt(0);
String strc=Character.toString(inputc);
System.out.println("String starting with character "+strc);
for(int i=0;i<list.size();i++)
{
if(list.get(i).startsWith(strc))
{
System.out.println(list.get(i));
}
}
}
public static void main(String args[])
{
ArrayL obj=new ArrayL();
obj.arraydisplay();
obj.appendatend();
obj.insertatpos();
obj.searchele();
obj.print();
}
}
```

OUTPUT:



```
ArrayList element are
[CSE, ISE, ME]

Enter the element to append at end
DS
[CSE, ISE, ME, DS]
Enter the position and element to insert
3
ECE
[CSE, ISE, ME, ECE, DS]

Enter the Array element to search
ISE
Element found at 1
Enter the starting charecter to print strings
C
String starting with character C
CSE
```

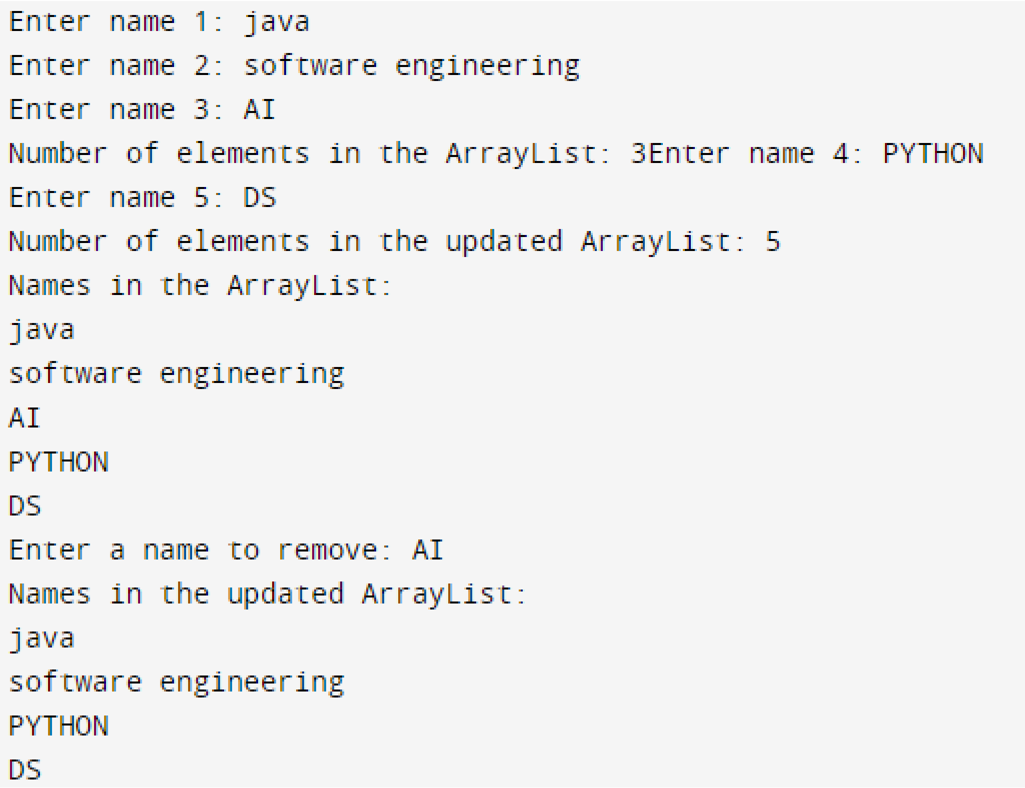
4) Create an ArrayList of type String and prompt that user for three names and add these names to your ArrayList.
b) Print a message with the number of elements in the ArrayList for the user using the size method.
c) Prompt the user for two more names and add them to the ArrayList and once again print a message with the number of elements in the ArrayList for the user.
d) Use a loop to print all of the names in the ArrayList for the user.
e) Ask the user for a name to remove, remove the value the user provides, and then use an enhanced for loop to print all of the names in the ArrayList for the user.

PROGRAM

```
import java.util.ArrayList;
import java.util.Scanner;
public class ArrayListExample {
    public static void main(String[] args) {
        // Create an ArrayList of type String
        ArrayList<String> namesList = new ArrayList<>();
        // Prompt the user for three names and add them to the ArrayList
        Scanner scanner = new Scanner(System.in);
        for (int i = 1; i <= 3; i++) {
            System.out.print("Enter name " + i + ": ");
            String name = scanner.nextLine();
            namesList.add(name);
        }
        // Print the number of elements in the ArrayList
        System.out.println("Number of elements in the ArrayList: " + namesList.size());
        // Prompt the user for two more names and add them to the ArrayList
        for (int i = 4; i <= 5; i++) {
            System.out.print("Enter name " + i + ": ");
            String name = scanner.nextLine();
            namesList.add(name);
        }
        // Print the number of elements in the updated ArrayList
        System.out.println("Number of elements in the updated ArrayList: " + namesList.size());
        // Use a loop to print all of the names in the ArrayList
        System.out.println("Names in the ArrayList:");
        for (String name : namesList) {
            System.out.println(name);
        }
        // Ask the user for a name to remove
        System.out.print("Enter a name to remove: ");
        String nameToRemove = scanner.nextLine();

        // Remove the name provided by the user
        namesList.remove(nameToRemove);
        // Use an enhanced for loop to print all of the names in the updated ArrayList
        System.out.println("Names in the updated ArrayList:");
        for (String name : namesList) {
        System.out.println(name);
        }
        // Close the scanner
        scanner.close();
    }
}
```

OUTPUT:



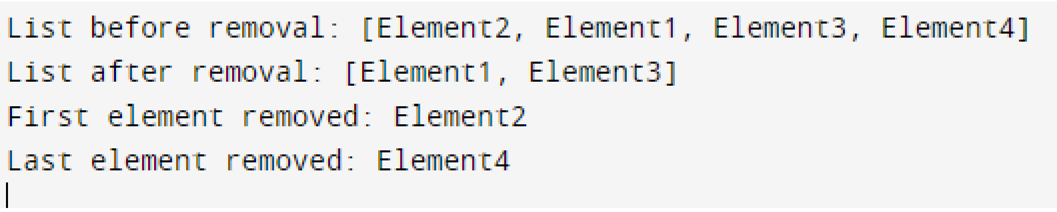
```
Enter name 1: java
Enter name 2: software engineering
Enter name 3: AI
Number of elements in the ArrayList: 3Enter name 4: PYTHON
Enter name 5: DS
Number of elements in the updated ArrayList: 5
Names in the ArrayList:
java
software engineering
AI
PYTHON
DS
Enter a name to remove: AI
Names in the updated ArrayList:
java
software engineering
PYTHON
DS
```

5) Write a Java program to add elements to the start of a list and to add elements to the end of the list. Obtain the first and last element. Remove first and last element.

PROGRAM

```
import java.util.ArrayList;
import java.util.List;
public class ListManipulation {
    public static void main(String[] args) {
        // Creating a List
        List<String> myList = new ArrayList<>();
        // Adding elements to the start of the list
        myList.add(0, "Element1");
        myList.add(0, "Element2");
        // Adding elements to the end of the list
        myList.add("Element3");
        myList.add("Element4");
        // Displaying the list before removal
        System.out.println("List before removal: " + myList);
        // Obtaining and removing the first element
        String firstElement = myList.remove(0);
        // Obtaining and removing the last element
        String lastElement = myList.remove(myList.size() - 1);
        // Displaying the list after removal
        System.out.println("List after removal: " + myList);
        // Displaying the first and last elements that were removed
        System.out.println("First element removed: " + firstElement);
        System.out.println("Last element removed: " + lastElement);
    }
}
```

OUTPUT:



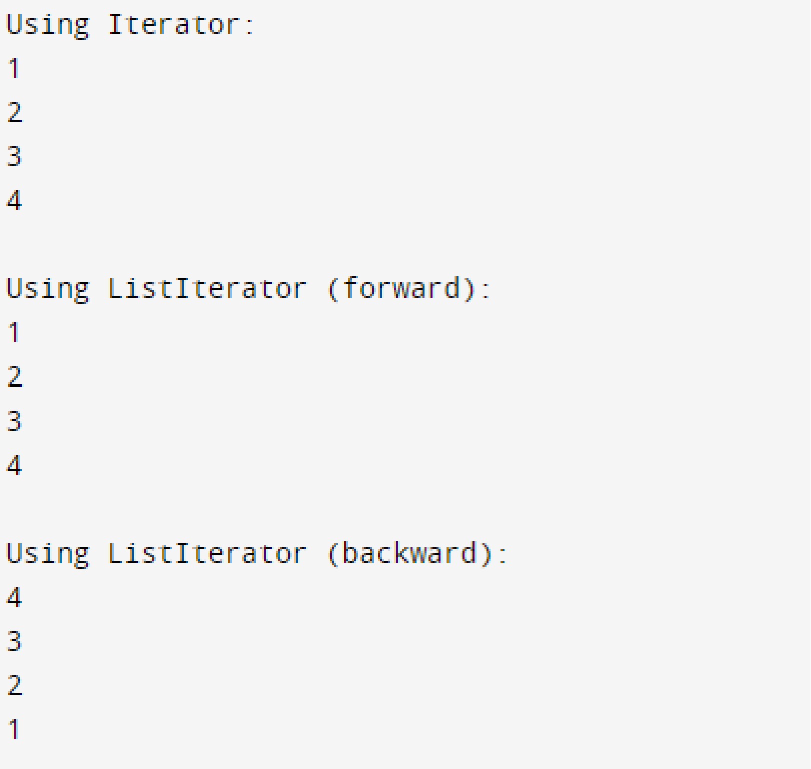
```
List before removal: [Element2, Element1, Element3, Element4]
List after removal: [Element1, Element3]
First element removed: Element2
Last element removed: Element4
```

6) Write a java program to implement both the Iterator and ListIterator interfaces.

PROGRAM

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.ListIterator;
public class IteratorExample {
    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<>();
        numbers.add(1);
        numbers.add(2);
        numbers.add(3);
        numbers.add(4);
        // Using Iterator
        System.out.println("Using Iterator:");
        Iterator<Integer> iterator = numbers.iterator();
        while (iterator.hasNext()) {
            System.out.println(iterator.next());
        }
        // Using ListIterator
        System.out.println("\nUsing ListIterator (forward):");
        ListIterator<Integer> listIterator = numbers.listIterator();
        while (listIterator.hasNext()) {
            System.out.println(listIterator.next());
        }
        // Using ListIterator in reverse
        System.out.println("\nUsing ListIterator (backward):");
        while (listIterator.hasPrevious()) {
            System.out.println(listIterator.previous());
        }
    }
}
```

OUTPUT:



```
Using Iterator:
1
2
3
4

Using ListIterator (forward):
1
2
3
4

Using ListIterator (backward):
4
3
2
1
```

7) Create a program to reads the data from two files and writes in to another file and to display only files in a specified location.

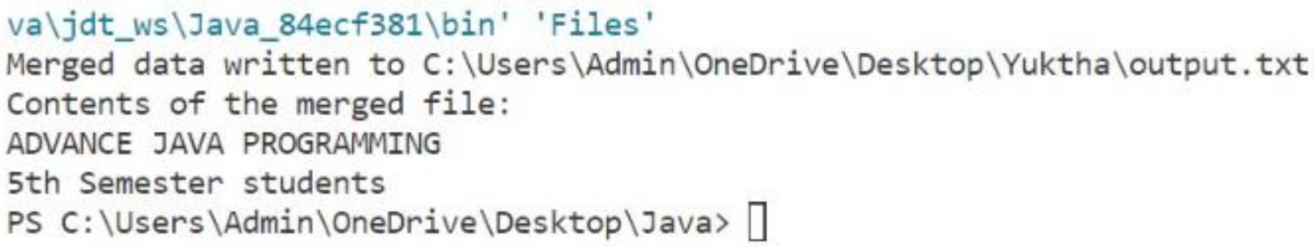
PROGRAM

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
public class MergeFilesAndDisplay {
 public static void main(String[] args) {
        // Specify the paths of the two input files
        String inputFile1 = "path/to/firstFile.txt";
        String inputFile2 = "path/to/secondFile.txt";
        // Specify the path of the output file
        String outputFile = "path/to/outputFile.txt";
        try {
            // Read data from the first file
            BufferedReader reader1 = new BufferedReader(new FileReader(inputFile1));
            String data1 = "";
            String line1;
            while ((line1 = reader1.readLine()) != null) {
                data1 += line1 + "\n";
            }
            reader1.close();
            // Read data from the second file
            BufferedReader reader2 = new BufferedReader(new FileReader(inputFile2));
            String data2 = "";
            String line2;
            while ((line2 = reader2.readLine()) != null) {
                data2 += line2 + "\n";
            }
            reader2.close();
            // Merge data from both files
            String mergedData = data1 + data2;
            // Write the merged data into the output file
            BufferedWriter writer = new BufferedWriter(new FileWriter(outputFile));
            writer.write(mergedData);
            writer.close();
            System.out.println("Merged data written to " + outputFile);
```

```
      // Display the contents of the output file
            System.out.println("Contents of the merged file:");
            BufferedReader mergedReader = new BufferedReader(new FileReader(outputFile));
            String line;
            while ((line = mergedReader.readLine()) != null) {
                System.out.println(line);
            }
            mergedReader.close();

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

OUTPUT:



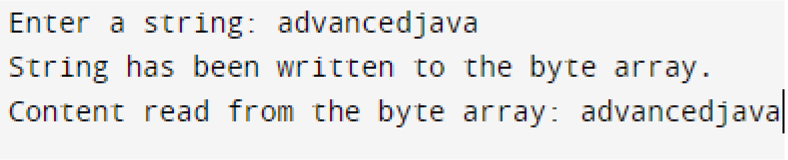
```
va\jdt_ws\Java_84ecf381\bin' 'Files'
Merged data written to C:\Users\Admin\OneDrive\Desktop\Yuktha\output.txt
Contents of the merged file:
ADVANCE JAVA PROGRAMMING
5th Semester students
PS C:\Users\Admin\OneDrive\Desktop\Java> ▯
```

8) Write a java program to read and write strings using byte array input and output streams

PROGRAM

```
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.Scanner;
public class ByteArrayExample {
    public static void main(String[] args) {
        // Read a string from the user
        String userInput = getUserInput();
        // Write the string to a byte array
        byte[] byteArray = writeToByteArray(userInput);
        // Read from the byte array and display content
        readAndDisplayFromByteArray(byteArray);
    }
    private static String getUserInput() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        return scanner.nextLine();
    }
    private static byte[] writeToByteArray(String inputString) {
        try (ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream()) {
            byte[] bytes = inputString.getBytes();
            byteArrayOutputStream.write(bytes);
            System.out.println("String has been written to the byte array.");
            return byteArrayOutputStream.toByteArray();
        } catch (IOException e) {
            System.out.println("An error occurred while writing to the byte array: " + e.getMessage());
            return new byte[0];
        }
    }
    private static void readAndDisplayFromByteArray(byte[] byteArray) {
        try (ByteArrayInputStream byteArrayInputStream = new ByteArrayInputStream(byteArray)) {
            byte[] buffer = new byte[1024];
            int bytesRead = byteArrayInputStream.read(buffer);
            String content = new String(buffer, 0, bytesRead);
            System.out.println("Content read from the byte array: " + content);
        } catch (IOException e) {
            System.out.println("An error occurred while reading from the byte array: " + e.getMessage());
    }
  }
}
```

OUTPUT:



```
Enter a string: advancedjava
String has been written to the byte array.
Content read from the byte array: advancedjava
```