Lab 6:-

Codes

1.

public class StringConstructorDemo {  
public static void main(String[] args) {  
String str1 = "Hello, World!"  
System.out.println("String created using a string literal: " + str1);  
}  
}  
public class StringConstructorDemo {  
public static void main(String[] args) {  
char[] charArray = {'H', 'e', 'l', 'l', 'o'};  
String str2 = new String(charArray);  
System.out.println("String created using the new keyword and char array: " + str2);  
}  
}  
public class StringConstructorDemo {  
public static void main(String[] args) {  
byte[] byteArray = {72, 101, 108, 108, 111};  
String str3 = new String(byteArray);  
System.out.println("String created using getBytes method: " + str3);  
}  
}  
public class StringConstructorDemo {  
public static void main(String[] args) {  
StringBuilder stringBuilder = new StringBuilder("Java");  
String str4 = new String(stringBuilder);  
System.out.println("String created using StringBuilder: " + str4);  
}  
}

2.

public class StringDemo {  
public static void main(String[] args) {

String exampleString = "Hello, World!"  
int length = exampleString.length();  
System.out.println("String Length: " + length);  
String stringLiteral1 = "Java"  
String stringLiteral2 = "Java" // Reusing the string literal  
System.out.println("String Literal 1: " + stringLiteral1);  
System.out.println("String Literal 2: " + stringLiteral2);  
System.out.println("Are String Literals Equal? " + (stringLiteral1 == stringLiteral2));  
String firstName = "John"  
String lastName = "Doe"  
String fullName = firstName + " " + lastName;  
System.out.println("Concatenated String: " + fullName);  
}  
}

3.

class Person {  
private String name;  
private int age;  
public Person(String name, int age) {  
[this.name](http://this.name/) = name;  
this.age = age;  
}  
@Override  
public String toString() {  
return "Person{name='" + name + "', age=" + age + '}';  
}  
}  
public class ToStringDemo {  
public static void main(String[] args) {  
Person person = new Person("John Doe", 25);  
System.out.println(person); // Output: Person{name='John Doe', age=25}  
}  
}

4.

public class SubstringExtraction {  
public static void main(String[] args) {  
String originalString = "Welcome to Bmsce college"  
char[] extractedChars = new char[5];  
originalString.getChars(11, 16, extractedChars, 0);  
String extractedString = new String(extractedChars);  
System.out.println("Extracted Substring: " + extractedString);  
}  
}

5.

public class GetBytesDemo {  
public static void main(String[] args) {  
String originalString = "Hello, World!"  
byte[] byteArray = originalString.getBytes();  
System.out.println("Byte Array: " + byteArray);  
System.out.print("Bytes: ");  
for (byte b : byteArray) {  
System.out.print(b + " ");  
}  
}  
}  
public class ToCharArrayDemo {  
public static void main(String[] args) {  
String originalString = "Java Programming"  
char[] charArray = originalString.toCharArray();  
System.out.println("Char Array: " + charArray);  
System.out.print("Chars: ");  
for (char c : charArray) {  
System.out.print(c + " ");  
}  
}  
}

6.

public class StringComparison {  
public static void main(String[] args) {String str1 = "Bmsce"  
String str2 = "College"  
String str3 = "BMSCE"  
System.out.println("Using equals(): Bmsce equals Bmsce -> " + str1.equals("Bmsce"));  
System.out.println("Using equals(): Bmsce equals College -> " + str1.equals(str2));  
System.out.println("Using equals(): Bmsce equals BMSCE -> " + str1.equals(str3));  
System.out.println("Using equalsIgnoreCase(): Bmsce equalsIgnoreCase BMSCE -> " + str1.equalsIgnoreCase(str3));  
}  
}

7.

public class RegionMatchesExample {  
public static void main(String[] args) {  
String mainString = "Welcome to Bmsce College of Engineering"  
String subString = "Bmsce College"  
boolean isMatched = mainString.regionMatches(11, subString, 0, subString.length());

if (isMatched) {  
System.out.println("Substring is matched.");  
} else {  
System.out.println("Substring is not matched.");  
}  
}  
}

8.

public class StartsWithExample {  
public static void main(String[] args) {  
String mainString = "Hello, World!"  
boolean startsWithHello = mainString.startsWith("Hello");  
boolean startsWithJava = mainString.startsWith("Java");  
System.out.println("Starts with 'Hello': " + startsWithHello); // Should be true  
System.out.println("Starts with 'Java': " + startsWithJava); // Should be false  
}  
}

9.

public class EndsWithExample {  
public static void main(String[] args) {  
String mainString = "Hello, World!"  
boolean endsWithWorld = mainString.endsWith("World!");  
boolean endsWithJava = mainString.endsWith("Java");  
System.out.println("Ends with 'World!': " + endsWithWorld); // Should be true  
System.out.println("Ends with 'Java': " + endsWithJava); // Should be false  
}  
}

10.

public class EqualsVsDoubleEquals {  
public static void main(String[] args) {  
String str1 = "Hello"  
String str2 = "Hello"  
String str3 = new String("Hello");  
boolean equalsResult1 = str1.equals(str2); // true  
boolean equalsResult2 = str1.equals(str3); // true  
boolean doubleEqualsResult1 = (str1 == str2); // true (due to string pooling)  
boolean doubleEqualsResult2 = (str1 == str3); // false (different objects)  
System.out.println("Using equals(): " + equalsResult1 + ", " + equalsResult2);  
System.out.println("Using ==: " + doubleEqualsResult1 + ", " + doubleEqualsResult2);  
}  
}

11.

import java.util.Arrays;

public class AlphabeticalSorting {

public static void main(String[] args) {

String[] words = {"van", "watch", "ball", "cat", "xmas", "yatch", "zee", "apple", "ice", "jug", "kite", "lift", "man", "net", "orange", "dog", "ent", "free", "gun", "hen", "parrot", "queen", "ring", "star", "tree", "umbrella"};

System.out.println("Original Array: " + Arrays.toString(words));

System.out.println("Sorted Array: " + Arrays.toString(words)); } }

12.

import java.util.Arrays;  
public class NumberSorting {  
public static void main(String[] args) {  
String[] numbers = {"10", "9", "8", "7", "6", "5", "4", "3", "2", "1"};  
Arrays.sort(numbers);  
System.out.println("Sorted Numbers:");  
for (String number : numbers) {  
System.out.println(number);  
}  
}  
}

13.

public class StringReplaceExample {  
public static void main(String[] args) {  
String originalString = "Thwas was a test. Thwas was, too.";

System.out.println("Original String: " + originalString);  
String targetSubstring = "was";  
String replacementString = "is";  
int index = originalString.indexOf(targetSubstring);  
StringBuilder result = new StringBuilder();

while (index != -1) {  
result.append(originalString.substring(0, index));  
result.append(replacementString);  
index += targetSubstring.length();  
originalString = originalString.substring(index);

index = originalString.indexOf(targetSubstring);  
}  
result.append(originalString);  
  
System.out.println("After Replacement: " + result.toString());  
}  
}

14.

public class StringConcatenationDemo {

public static void main(String[] args) {

String s1 = "hello";

String s2 = "world"; //

String result = s1.concat(s2);

System.out.println("Concatenated String: " + result); } }

15.

public class StringReplaceDemo {

    public static void main(String[] args) {  
        String originalString = "Welcome to College";  
        String modifiedString = originalString.replace("College", "Commege");  
        System.out.println("Original String: " + originalString);  
        System.out.println("Modified String: " + modifiedString);  
    }

16.

public class StringTrimDemo {  
    public static void main(String[] args) {  
        String originalString = " Hello Friends ";  
  
        String trimmedString = originalString.trim();  
  
        System.out.println("Original String: '" + originalString + "'");  
        System.out.println("Trimmed String: '" + trimmedString + "'");  
    }  
}

17.

import java.util.Arrays;

import java.util.Comparator;

import java.util.Scanner;

class Student {

private int registrationNumber;

private String fullName;

private short semester;

private float cgpa;

public Student() {

}

public Student(int registrationNumber, String fullName, short semester, float cgpa) {

this.registrationNumber = registrationNumber;

this.fullName = fullName;

this.semester = semester;

this.cgpa = cgpa;

} public void display() {

System.out.println("Registration Number: " + registrationNumber);

System.out.println("Full Name: " + fullName);

System.out.println("Semester: " + semester);

System.out.println("CGPA: " + cgpa);

System.out.println("------------------------");

}

public int getRegistrationNumber() {

return registrationNumber;

}

public String getFullName() {

return fullName;

}

public short getSemester() {

return semester;

}

public float getCgpa() {

return cgpa;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Student[] students = new Student[5];

for (int i = 0; i < 5; i++) {

System.out.println("Enter details for student " + (i + 1) + ":");

System.out.print("Registration Number: ");

int regNumber = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Full Name: ");

String name = scanner.nextLine();

System.out.print("Semester: ");

short semester = scanner.nextShort();

System.out.print("CGPA: ");

float cgpa = scanner.nextFloat();

students[i] = new Student(regNumber, name, semester, cgpa);

}

System.out.println("\nStudent Records:");

for (Student student : students) {

student.display();

}

sortByCGPA(students);

System.out.println("\nStudent Records (Sorted by CGPA):");

for (Student student : students) {

student.display();

}

sortByName(students);

System.out.println("\nStudent Records (Sorted by Name):");

for (Student student : students) {

student.display();

}

}

private static void sortByCGPA(Student[] students) {

Arrays.sort(students, Comparator.comparingDouble(Student::getCgpa).reversed());

}

private static void sortByName(Student[] students) {

Arrays.sort(students, Comparator.comparing(Student::getFullName));

}

}

18.

public class StringBufferDemo {

public static void main(String[] args) {

StringBuffer stringBuffer = new StringBuffer("Hello, World!");

stringBuffer.setLength(5);

System.out.println("After setLength(5): " + stringBuffer);

char charAtIndex = stringBuffer.charAt(1);

System.out.println("charAt(1): " + charAtIndex);

stringBuffer.setCharAt(1, 'a');

System.out.println("After setCharAt(1, 'a'): " + stringBuffer);

char[] charArray = new char[5];

stringBuffer.getChars(0, 5, charArray, 0);

System.out.println("getChars(0, 5, charArray, 0): " + new String(charArray));

}

}

public class StringBufferDemo2 {

public static void main(String[] args) {

StringBuffer stringBuffer = new StringBuffer("Hello");

stringBuffer.append(" World!");

System.out.println("After append: " + stringBuffer);

stringBuffer.insert(6, ", Java");

System.out.println("After insert(6, \", Java\"): " + stringBuffer);

stringBuffer.reverse();

System.out.println("After reverse: " + stringBuffer);

stringBuffer.delete(5, 11);

System.out.println("After delete(5, 11): " + stringBuffer);

stringBuffer.deleteCharAt(1);

System.out.println("After deleteCharAt(1): " + stringBuffer);

}

}public class StringBufferDemo3 { public static void main(String[] args) { StringBuffer stringBuffer = new StringBuffer("Hello, Java!"); // replace() method stringBuffer.replace(7, 12, "World"); System.out.println("After replace(7, 12, \"World\"): "

19.

abstract class Bird {  
    public abstract void fly();  
    public abstract void makeSound();  
}  
class Eagle extends Bird {  
    @Override  
    public void fly() {  
        System.out.println("Eagle flies high in the sky.");  
    }  
    @Override  
    public void makeSound() {  
        System.out.println("Eagle screeches loudly.");  
    }  
}  
class Hawk extends Bird {  
    @Override  
    public void fly() {  
        System.out.println("Hawk soars gracefully through the air.");  
    }  
    @Override  
    public void makeSound() {  
        System.out.println("Hawk emits a piercing cry.");  
    }  
}  
public class BirdTest {  
    public static void main(String[] args) {  
        // Create instances of Eagle and Hawk  
        Eagle eagle = new Eagle();  
        Hawk hawk = new Hawk();  
        System.out.println("Eagle:");  
        eagle.fly();  
        eagle.makeSound();  
        System.out.println();  
        System.out.println("Hawk:");  
        hawk.fly();  
        hawk.makeSound();  
    }  
}

20.

abstract class Shape {

public abstract double calculateArea();

public abstract double calculatePerimeter();

}

class Circle extends Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double calculateArea() {

return Math.PI \* radius \* radius;

}

@Override

public double calculatePerimeter() {

return 2 \* Math.PI \* radius;

}

}

class Triangle extends Shape {

private double side1, side2, side3;

public Triangle(double side1, double side2, double side3) {

this.side1 = side1;

this.side2 = side2;

this.side3 = side3;

}

@Override

public double calculateArea() {

triangle

double s = (side1 + side2 + side3) / 2.0;

return Math.sqrt(s \* (s - side1) \* (s - side2) \* (s - side3));

}

@Override

public double calculatePerimeter() {

return side1 + side2 + side3;

}

}

public class ShapeTest {

public static void main(String[] args) {

// Create instances of Circle and Triangle

Circle circle = new Circle(5.0);

Triangle triangle = new Triangle(3.0, 4.0, 5.0);

for Circle

System.out.println("Circle:");

System.out.println("Area: " + circle.calculateArea());

System.out.println("Perimeter: " + circle.calculatePerimeter());

System.out.println();

for Triangle

System.out.println("Triangle:");

System.out.println("Area: " + triangle.calculateArea());

System.out.println("Perimeter: " + triangle.calculatePerimeter());

}

}

Observation book :-







