

19. Write a Java program to create an abstract class Bird with abstract methods fly() and makeSound(). Create subclasses Eagle and Hawk that extend the Bird class and implement the respective methods to describe how each bird flies and makes a sound.

```
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.");
    }
    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.");
    }
    public void makeSound() {
        System.out.println("Hawk emits a piercing cry.");
    }
}
public class BirdTest {
    public static void main(String[] args) {
        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();
    }
}
```

```
java -cp /tmp/5b405VteG4 BirdTest
Eagle:Eagle flies high in the sky.
Eagle screeches loudly.

Hawk:
Hawk soars gracefully through the air.
Hawk emits a piercing cry.
```

17. Design a class which represents a student. Every student record is made up of the following fields. i) Registration number (int) ii) Full Name (String) iii) Semester (short) iv) CGPA (float)

Write member functions to do the following.

- a) Provide default and parameterized constructors to this class
- b) Write display method which displays the record. Test the class by writing suitable main method.
- c) Create an array of student record to store minimum of 5 records in it. Input the records and display them.
- d. Perform the following operations by adding member functions to the program implemented in the above question i) Sort the student records with respect to CGPA. ii) Sort the student record with respect to name.

```
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[2];
        for (int i = 0; i < 2; 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));
    }
}

```

```

Output
java -cp /tmp/AQFKpf2hJn Student
Enter details for student 1:
Registration Number: 01
Full Name: vinu
Semester: 3
CGPA: 9.8
Enter details for student 2:
Registration Number: 02
Full Name: pooja
Semester: 3
CGPA: 10
Student Records:
Registration Number: 1
Full Name: vinu
Semester: 3
CGPA: 9.8
-----
Registration Number: 2
Full Name: pooja
Semester: 3
CGPA: 10.0
Student Records (Sorted by CGPA):
Registration Number: 2 Full Name: pooja Semester: 3 CGPA
: 10.0 -----
Registration Number: 1 Full Name: vinu Semester: 3 CGPA
: 9.8 -----

```

20. Write a Java program to create an abstract class Shape with abstract methods calculateArea() and calculatePerimeter(). Create subclasses Circle and Triangle that extend the Shape class and implement the respective methods to calculate the area and perimeter of each shape

```

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;
    }
    public double calculateArea() {
        return Math.PI * radius * radius;
    }
    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;
    }
    public double calculateArea() {
        double s = (side1 + side2 + side3) / 2.0;
        return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));
    }
    public double calculatePerimeter() {
        return side1 + side2 + side3;
    }
}
public class ShapeTest {
    public static void main(String[] args) {
        Circle circle = new Circle(5.0);
        Triangle triangle = new Triangle(3.0, 4.0, 5.0);
        System.out.println("Circle:");
        System.out.println("Area: " + circle.calculateArea());
        System.out.println("Perimeter: " + circle.calculatePerimeter());
        System.out.println();
        System.out.println("Triangle:");
        System.out.println("Area: " + triangle.calculateArea());
        System.out.println("Perimeter: " + triangle.calculatePerimeter());
    }
}

```

```
Output
java -cp /tmp/5b405VteG4 ShapeTest
Circle:
Area: 78.53981633974483Perimeter: 31.41592653589793

Triangle:
Area: 6.0
Perimeter: 12.0
```

18. Demonstrate string buffer functions like Setlength(), Charat(), setcharat() , getchars() ,append() ,Insert(), reverse(),delete(),deletecharat(), Replace(),substring() with simple java programs.

```
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);
    }
}

```

Output

```

java -cp /tmp/5b405VteG4 StringBufferDemo
After setLength(5): Hello
charAt(1): eAfter setCharAt(1, 'a'): HallogetChars(0, 5, charArray, 0): Hallo

```

Output

```

java -cp /tmp/5b405VteG4 StringBufferDemo2
After append: Hello World!After insert(6, ", Java"): Hello , JavaWorld!
After reverse: !dlrowaW , olleH
After delete(5, 11): !dlro, olleH
After deleteCharAt(1): !lro, olleH
|

```

Output

```

java -cp /tmp/5b405VteG4 StringBufferDemo3
After replace(7, 12, "World"): Hello, WorldSubstring(7, 12): World

```

16. Write a java program to demonstrate trim() for " Hello Friends "

```

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 + "");
    }
}

```



## Output

```
java -cp /tmp/5b405VteG4 StringTrimDemo  
Original String: ' Hello Friends '  
Trimmed String: 'Hello Friends'
```

15. Write a java program to demonstrate replace(). Replace "College" with "Commege"

```
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);  
    }  
}
```

```
java -cp /tmp/5b405VteG4 StringReplaceDemo  
Original String: Welcome to College  
Modified String: Welcome to Commege  
|
```

14. Write a java program to demonstrate concat() for s1="hello" and s2="world"

```
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);  
    }  
}
```

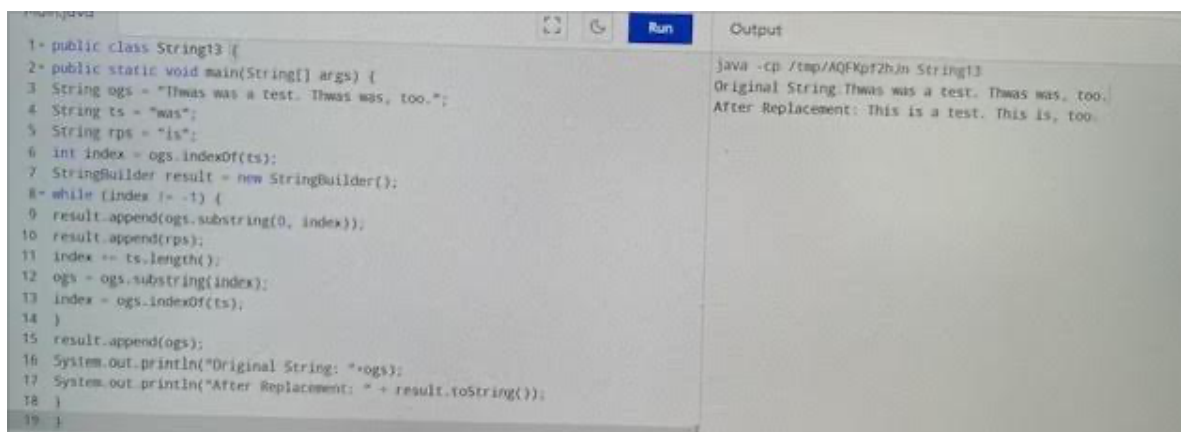


```
java -cp /tmp/5b405VteG4 StringConcatenationDemo
Concatenated String: helloworld
```

13. Write a Java program using `substring()` , `indexOf()`, `+` , for replacing "was" to "is"

"Thwas was a test. Thwas was, too."

```
public class StringReplaceExample {
    public static void main(String[] args) {
        String originalString = "Thwas was a test. Thwas was, too."
        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("Original String: " + originalString);
        System.out.println("After Replacement: " + result.toString());
    }
}
```



The screenshot shows an IDE window with a file named `String13.java`. The code in the editor is as follows:

```
1. public class String13 {
2.     public static void main(String[] args) {
3.         String ogs = "Thwas was a test. Thwas was, too.";
4.         String ts = "was";
5.         String rps = "is";
6.         int index = ogs.indexOf(ts);
7.         StringBuilder result = new StringBuilder();
8.         while (index != -1) {
9.             result.append(ogs.substring(0, index));
10.            result.append(rps);
11.            index += ts.length();
12.            ogs = ogs.substring(index);
13.            index = ogs.indexOf(ts);
14.        }
15.        result.append(ogs);
16.        System.out.println("Original String: " + ogs);
17.        System.out.println("After Replacement: " + result.toString());
18.    }
19. }
```

The `Run` button is highlighted. The `Output` pane on the right shows the following text:

```
java -cp /tmp/AQFKpt2hjn String13
Original String:Thwas was a test. Thwas was, too.
After Replacement: This is a test. This is, too.
```

9. Demonstrate `endsWith()` to give output true and false.

```
public class EndsWithExample {
    public static void main(String[] args) {
        String mainString = "Hello, Buddy!"
        boolean endsWithWorld = mainString.endsWith("Buddy!");
```

```

boolean endsWithJava = mainString.endsWith("Vinuthna");
System.out.println("Ends with 'Buddy!': " + endsWithWorld);
System.out.println("Ends with 'Vinuthna': " + endsWithJava);
}
}

```

The screenshot shows an IDE with a Java file named Stringq9.java. The code defines a class Stringq9 with a main method. It initializes a String mainString to "Hello, Buddy!". It then checks if mainString ends with "Buddy!" and "Java". The output window shows the results: "Ends with Buddy!: true" and "Ends with vinuthna: false".

```

1- public class Stringq9 {
2- public static void main(String[] args) {
3
4 String mainString = "Hello, Buddy!";
5
6
7 boolean endsWithWorld = mainString.endsWith("Buddy!");
8 boolean endsWithJava = mainString.endsWith("Java");
9
10 System.out.println("Ends with Buddy!: " + endsWithWorld);
11 System.out.println("Ends with vinuthna: " + endsWithJava);
12 }
13 }

```

Output

```

java -cp /tmp/AQFKpf2hjn Stringq9
Ends with Buddy!: true
Ends with vinuthna: false

```

12. Write a Java program to perform sorting of numbers from 10 to 1 using compareto()

```

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);
}
}
}

```

The screenshot shows an IDE with a Java file named Stringt2.java. The code imports java.util.Arrays and defines a class Stringt2 with a main method. It initializes a String[] numbers array with values from "10" down to "1". It then sorts the array using Arrays.sort() and prints each element. The output window shows the sorted numbers: "11023456789".

```

1- import java.util.Arrays;
2
3- public class Stringt2 {
4- public static void main(String[] args) {
5
6 String[] numbers = {"10", "9", "8", "7", "6", "5", "4", "3", "2", "1"};
7
8 Arrays.sort(numbers);
9
10 System.out.println("Sorted Numbers:");
11 for (String number : numbers) {
12 System.out.println(number);
13 }
14 }
15 }

```

Output

```

java -cp /tmp/AQFKpf2hjn Stringt2
Sorted Numbers:11023456789

```

11. Write a java program to perform sorting for alphabets using compareto()  
"van", "watch","ball", "cat","xmas","yatch"," zee",  
apple","ice","jug","kite","lift","man","net","orange","dog","ent","free","gun","hen",  
parrot","q  
ueen","ring","star","tree","umbrella"

```

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"};
    Arrays.sort(words);
    System.out.println("Sorted Words:");
    for (String word : words) {
        System.out.println(word);
    }
}

```

The screenshot shows an IDE window titled 'Main.java' with the following code:

```

1- import java.util Arrays;
2
3- public class Stringts {
4-     public static void main(String[] args) {
5
6-         String[] words = {"van", "watch", "ball", "cat", "xmas", "yatch", "zee", "apple", "free",
7-             "ice", "jug", "kite",
8-             "lift", "man", "net", "orange", "dog", "ent", "free", "gun", "hen", "parrot",
9-             "queen",
10            "ring", "star", "tree", "umbrella"};
11
12            Arrays.sort(words);
13
14            System.out.println("Sorted words:");
15            for (String word : words) {
16                System.out.println(word);
17            }
18        }
19    }

```

To the right of the code editor is an 'Output' pane displaying the sorted words in alphabetical order:

```

apple
ball
cat
dog
ent
free
gun
hen
ice
jug
kite
lift
man
net
orange
parrot
queen
ring
star
tree
umbrella
van
watch
xmas
yatch

```

```

1- import java.util.Arrays;
2-
3- public class StringIt {
4-     public static void main(String[] args) {
5-
6-         String[] words = {"van", "watch", "ball", "cat", "amas", "yatch", "zee", "apple",
7-             "ice", "jug", "kite",
8-             "lift", "man", "net", "orange", "dog", "ent", "free", "gun", "hen", "parrot",
9-             "queen",
10            "ring", "star", "tree", "umbrella"};
11
12            Arrays.sort(words);
13
14            System.out.println("Sorted Words:");
15            for (String word : words) {
16                System.out.println(word);
17            }
18        }
19    }

```

Output

```

java -cp /tmp/AQ8KpT2hJm StringIt
Sorted Words:
apple
ball
cat
dog
ent
free
gun
hen
ice
jug
kite
lift
man
net
orange
parrot
queen
ring
star
tree
umbrella
van
watch
amas
yatch

```

10. Demonstrate a java program to show the output for equals() versus ==

```

public class EqualsVsDoubleEquals {
    public static void main(String[] args) {
        // Given strings
        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);
    }
}

```

```

1- class StringTo {
2-
3-     public static void main(String args[]) {
4-
5-         String s1 = "Hello";
6-
7-         String s2 = new String(s1);
8-
9-         System.out.println(s1 + " equals " + s2 + " -> " +
10            s1.equals(s2));
11
12            System.out.println(s1 + " == " + s2 + " -> " + (s1 == s2));
13        }
14    }

```

Output

```

java -cp /tmp/AQ8KpT2hJm StringTo
Hello equals Hello -> true
Hello == Hello -> false

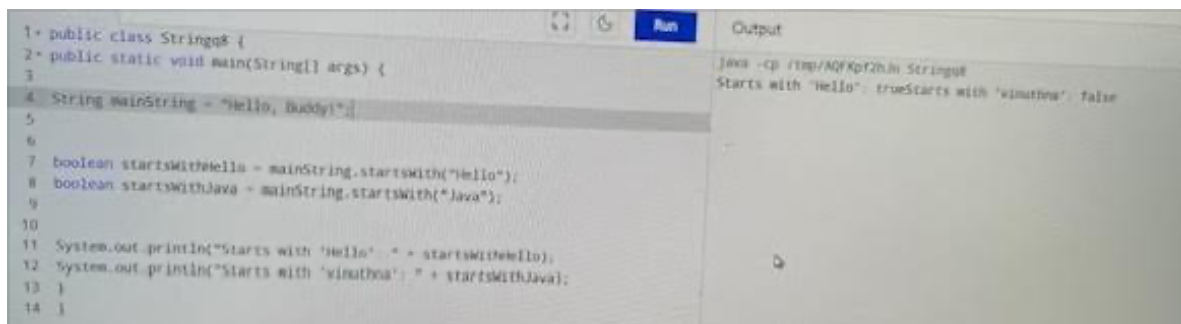
```

8. Demonstrate startwith() to give output true and false.

```

public class StartsWithExample {
public static void main(String[] args) {
String mainString = "Hello, Buddy!"
boolean startsWithHello = mainString.startsWith("Hello");
boolean startsWithvinuthna = mainString.startsWith("Vinuthna");
System.out.println("Starts with 'Hello': " + startsWithHello);
System.out.println("Starts with 'Vinuthna': " + startsWithJvinuthna);
}
}

```



The screenshot shows an IDE with a Java class named `String8`. The code defines a `main` method that takes a `String[] args` parameter. It sets `mainString` to "Hello, Buddy!". It then checks if `mainString` starts with "Hello" and "Vinuthna" using the `startsWith` method. The output window shows the results: "Starts with 'Hello': true" and "Starts with 'vinuthna': false".

```

1 public class String8 {
2 public static void main(String[] args) {
3
4 String mainString = "Hello, Buddy!";
5
6
7 boolean startsWithHello = mainString.startsWith("Hello");
8 boolean startsWithJava = mainString.startsWith("Java");
9
10
11 System.out.println("Starts with 'Hello': " + startsWithHello);
12 System.out.println("Starts with 'vinuthna': " + startsWithJava);
13 }
14 }

```

Output:

```

java -cp /tmp/NQFKpT2hJn String8
Starts with 'Hello': trueStarts with 'vinuthna': false

```

7. Using `regionMatches()` find the substring "Bmsce college " from the string "Welcome to Bmsce College of Engineering" , if matches display substring is matched otherwise display not matched

```

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.");
}
}
}
}

```

```

1- public class Stringq7 {
2- public static void main(String[] args) {
3
4 String mainString = "welcome to Bmsce College of Engineering";
5 String subString = "Bmsce College";
6
7 boolean isMatched = mainString.regionMatches(11, subString, 0, subString.length
8   ());
9
10- if (isMatched) {
11 System.out.println("Substring is matched.");
12- } else {
13 System.out.println("Substring is not matched.");
14 }
15 }
16 }

```

Output

```

java -cp /tmp/AQ/Rpt2hJn Stringq7
SubString is matched.

```

6. Check the following output and write the java programs using string function

Bmsce equals Bmsce -> true

Bmsce equals College -> false

Bmsce equals BMSCE -> false

Bmsce equalsIgnoreCase BMSCE -> true

```
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));
```

```
}
```

```
}
```

```

Main.java
1- public class Stringq6 {
2- public static void main(String[] args) {
3
4 String str1 = "Bmsce";
5 String str2 = "College";
6 String str3 = "BMSCE";
7
8
9 System.out.println("Using equals(): Bmsce equals Bmsce -> " + str1.equals("Bmsce"
10   ));
11 System.out.println("Using equals(): Bmsce equals College -> " + str1.equals(str2
12   ));
13 System.out.println("Using equals(): Bmsce equals BMSCE -> " + str1.equals(str3));
14
15 System.out.println("Using equalsIgnoreCase(): Bmsce equalsIgnoreCase BMSCE -> " +
16   str1.equalsIgnoreCase(str3));
17 }
18 }

```

Output

```

java -cp /tmp/AQ/Rpt2hJn Stringq6
Using equals(): Bmsce equals Bmsce -> true
Using equals(): Bmsce equals College -> false
Using equals(): Bmsce equals BMSCE -> false
Using equalsIgnoreCase(): Bmsce equalsIgnoreCase BMSCE -> true

```

5. Demonstrate getbytes(), toCharArray() with proper java programs

```
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 + " ");
        }
    }
}

```

The screenshot shows an IDE with a code editor on the left and an output window on the right. The code in the editor is for a class named `ToCharArrayQ52`. It defines a `main` method that takes a `String[] args` parameter. Inside the method, a `String` variable `originalString` is assigned the value `"in java lab"`. A `char[]` variable `charArray` is created by calling `originalString.toCharArray()`. The code then prints the `Char Array` and the individual `Chars` of the array. The output window on the right shows the command `java -cp /tmp/AQfKpf2HJn ToCharArrayQ52` and the resulting output: `Char Array: {C83cd1f1c8}` and `Chars: i n j a v a l a b |`.

The screenshot shows an IDE with a code editor on the left and an output window on the right. The code in the editor is for a class named `GetBytesQ51`. It defines a `main` method that takes a `String[] args` parameter. Inside the method, a `String` variable `ogs` is assigned the value `"Hello, world!"`. A `byte[]` variable `byteArray` is created by calling `ogs.getBytes()`. The code then prints the `Byte Array` and the individual `Bytes` of the array. The output window on the right shows the command `java -cp /tmp/AQfKpf2HJn GetBytesQ51` and the resulting output: `Byte Array: {882dbcf6b2}` and `Bytes: 72 101 108 108 111 44 32 87 111 116 104 100 33`.

4. Using `getchars()` , extract Bmsce from " Welcome to Bmsce college"

```

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);
}
}

```

```

1 public class Sseq4 {
2     public static void main(String[] args) {
3
4         String originalString = "Welcome to Basce college";
5
6         char[] extractedChars = new char[5];
7         originalString.getChars(11, 16, extractedChars, 0);
8
9         String extractedString = new String(extractedChars);
10
11        System.out.println("Extracted Substring: " + extractedString);
12    }
13 }
14

```

Output

```

java -cp /tmp/AQFRpf2h In Sseq4
Extracted Substring: Basce

```

### 3.Demonstrate toString()

```

class Person {
    private String name;
    private int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    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);
    }
}

```

```

1 class Box {
2     double width;
3     double height;
4     double depth;
5     Box(double w, double h, double d) {
6         width = w;
7         height = h;
8         depth = d;
9     }
10    public String toString() {
11        return "Dimensions are " + width + " by " +
12            depth + " by " + height + " ";
13    }
14 }

15 class ToStringDemo {
16     public static void main(String args[]) {
17         Box b = new Box(10, 12, 14);
18         String s = "Box b: " + b; // demonstrate Box object
19         System.out.println(b); // convert box to string
20         System.out.println(s);
21     }
22 }

```

Output

```

java -cp /tmp/AQFRpf2h In ToStringDemo
Dimensions are 10.0 by 14.0 by 12.0
Box b: Dimensions are 10.0 by 14.0 by 12.0

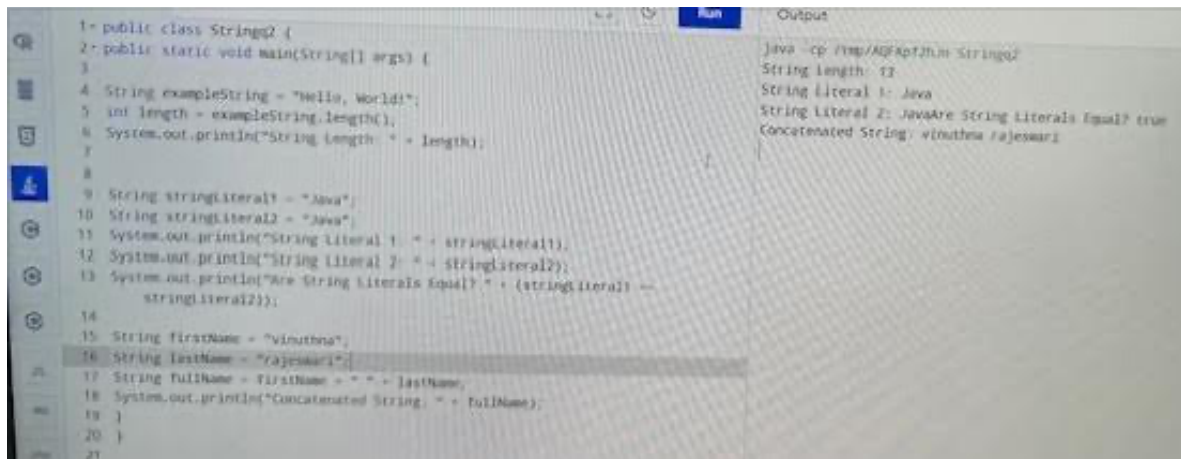
```

### 2.Demonstrate string length, string literal, string concat

```

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"
        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 = "Vinuthna"
        String lastName = "rajeswari"
        String fullName = firstName + " " + lastName;
        System.out.println("Concatenated String: " + fullName);
    }
}

```



The screenshot shows an IDE with a Java file named `String2.java`. The code is identical to the one in the first block. The output window on the right shows the following results:

```

Java - cp /tmp/AGFpTJhM String2
String length: 12
String Literal 1: Java
String Literal 2: Java
Are String Literals Equal? true
Concatenated String: Vinuthna rajeswari

```

1.Demonstrate various string constructor with proper java programs.

```

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);
}
}

```

```

Main.java
1 public class String4 {
2 public static void main(String[] args) {
3   StringBuilder stringBuilder = new StringBuilder("Java");
4   String str4 = new String(stringBuilder);
5   System.out.println(str4);
6 }
7 }
8

```

Run

Output

```

java -cp /tmp/AQFKpf2hJn String4
Java

```

```

Main.java
1 public class String3 {
2 public static void main(String[] args) {
3   byte[] byteArray = {72, 101, 108, 108, 111};
4   String str3 = new String(byteArray);
5   System.out.println(str3);
6 }
7 }
8

```

Run

Output

```

java -cp /tmp/AQFKpf2hJn String3
Hello

```

```

Main.java
1 public class String2 {
2 public static void main(String[] args) {
3   char[] charArray = {'H', 'e', 'l', 'l', 'o'};
4   String str2 = new String(charArray);
5   System.out.println(str2);
6 }
7 }

```

Run

Output

```

java -cp /tmp/AQFKpf2hJn String2
Hello

```

```

Main.java
1 class Main {
2 public static void main(String[] args) {
3   String str1 = "Hello, World!";
4   System.out.println(str1);
5 }
6 }

```

Run

Output

```

java -cp /tmp/AQFKpf2hJn Main
Hello, World!

```

