Jan/16/24

Lab - 6

1) Demonstrate various constructors in String

Str 1 : Hello
Str 2 : world
Str 3 : Or !
Str 4 : Hello
Str 5 : Java

2)
Str = "Hello. World"
String length = 12

3) demonstrate toString()

output:

101. mrunalini Bangalore
102. mrunal Mysore.

4) Using getchars(), Extract Bmscc from "welcome to Bmscc college"

Output : Bmscc

5) Demonstrate getbytes(), tochar Array()
with Proper Java Programs

Output : (i) getbytes()
Hello, world!

(ii) tochar Array()
J a v a P r o g r a m m i n g

6) Check the following output and write the Java Programs using Stringfunction

i. Bmscc Equals Bmscc → true
ii. Bmscc Equals College → false

Bmscc Equals BMSCE → false.

Bmscc Equals IgnoreCase BMSCE → true

Output :-

```
Public class StringComparisonExample {
    Public Static Void main (String [] args) {
        String Str1 = "Bmscc";
        String Str2 = "College";
        String Str3 = "BMSCE";
System.out.Println (Str1 + "Equals" + Str1 +
        "->" + Str1.Equals (Str1));
System.out.Println (Str2 + "Equals" + Str2 +
        "->" + Str1.Equals (Str2));
System.out.Println (Str1 + "Equals" + Str3 +
        "->" + Str1.Equals (Str3));
System.out.Println (Str1 + "Equals" + Str4 +
        "->" + Str1.Equals (Str4));
    }
}
```

7) Using regionMatches() find Substring "Bmscc College" from String "Welcome to Bmscc College of Engineering", display Substring is matched otherwise display not matched

Output :- Substring is matched

8) Demonstrate StartsWith() to give output true & false.

Sol:- import java.lang.String;

Public class Star {

Public Static Void main

output: Check whether String Starts with Welcome at Pos 11: false

Check whether String Starts With Bmscc at Pos 11: ~~false~~ true.

9) Demonstrate EndsWith() to give output ~~true~~ and false

Output: True
False

10) Demonstrate a java Program to show output for Equals() vs ==

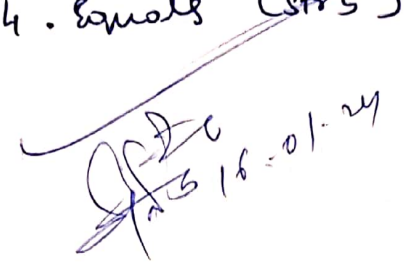
Output: Using == with String literals:
Str1 == Str2 : true
Str1 == Str3 : false
Using Equals() with String literals:
Str1.Equals(Str2) : true
Str1.Equals(Str3) : true

using == with new String Objects:

Str4 == Str5 : false

using equals() with new String Objects:

Str4.equals(Str5): true



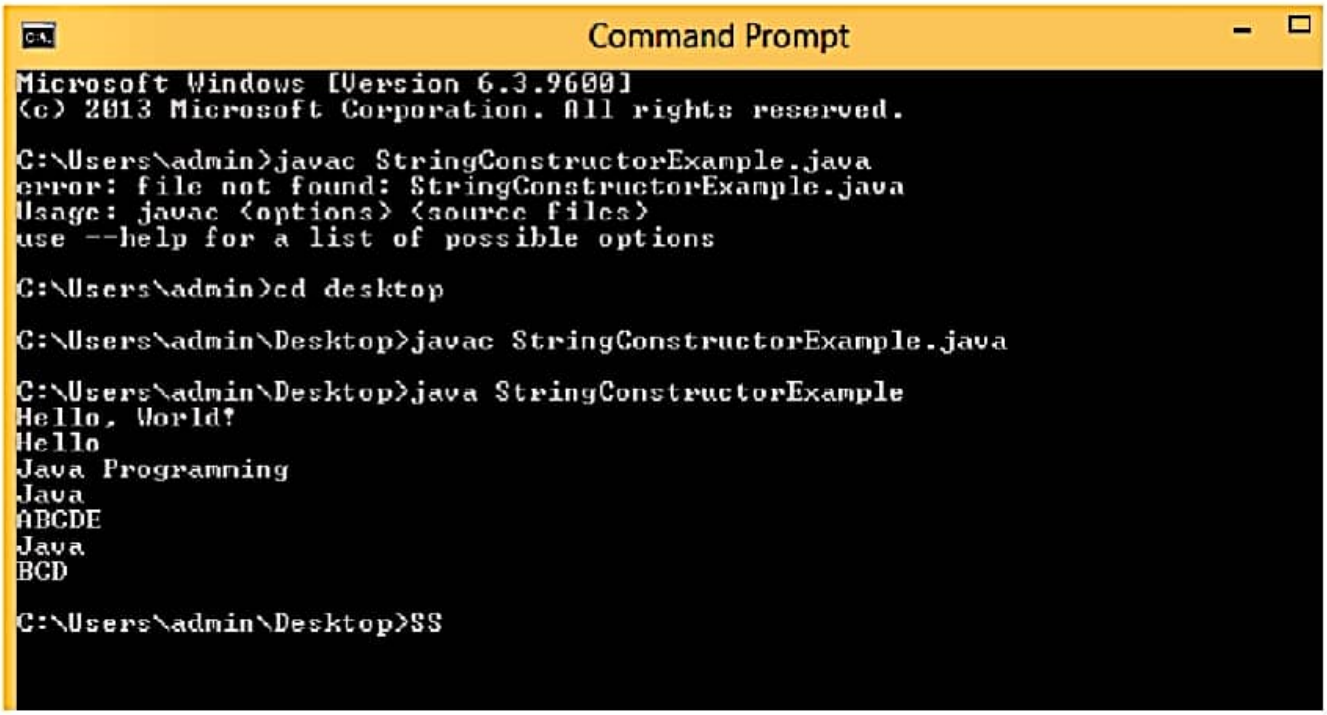
LAB -6

PROGRAM-1

SOURCE CODE:

```
public class StringConstructorExample {
   public static void main(String[] args) {
      // Using String Literal
      String str1 = "Hello, World!";

      // Using new keyword and character array
      char[] charArray = {'H', 'e', 'l', 'l', 'o'};
      String str2 = new String(charArray);

      // Using another String
      String original = "Java Programming";
      String str3 = new String(original);

      // Using StringBuilder
      StringBuilder stringBuilder = new StringBuilder("Java");
      String str4 = new String(stringBuilder);

      // Using byte array
      byte[] byteArray = {65, 66, 67, 68, 69}; // ASCII values for A, B, C, D, E
      String str5 = new String(byteArray);

      // Using part of a character array
      char[] charArray2 = {'J', 'a', 'v', 'a', ' ', 'P', 'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g'};
      String str6 = new String(charArray2, 0, 4); // "Java"

      // Using part of a byte array
      byte[] byteArray2 = {65, 66, 67, 68, 69}; // ASCII values for A, B, C, D, E
      String str7 = new String(byteArray2, 1, 3); // "BCD"

      // Print the strings
      System.out.println(str1);
      System.out.println(str2);
      System.out.println(str3);
      System.out.println(str4);
      System.out.println(str5);
      System.out.println(str6);
      System.out.println(str7);
   }
}
```

OUTPUT:



PROGRAM-2

SOURCE CODE:

```
public class StringOperationsDemo {
   public static void main(String[] args) {
      // String Literal
      String str1 = "Hello, ";
      String str2 = "Java!";

      // String Concatenation
      String result = str1 + str2;

      // String Length
      int length = result.length();

      // Displaying the results
      System.out.println("String 1: " + str1);
      System.out.println("String 2: " + str2);
      System.out.println("Concatenated String: " + result);
      System.out.println("Length of Concatenated String: " + length);
   }
}
```

OUTPUT:

```
C:\Users\Admin\Desktop>javac StringOperationsDemo.java

C:\Users\Admin\Desktop>java StringOperationsDemo
String 1: Hello,
String 2: Java!
Concatenated String: Hello, Java!
Length of Concatenated String: 12
```

PROGRAM-3 AND 4

SOURCE CODE:

```
public class College {
    private String name;


    public College(String name) {
        this.name = name;
    }



    public String toString() {
        return "College{name='" + name + "'}";
    }


    public void extractSubstring() {
        char[] targetArray = new char[5];

        name.getChars(11, 16, targetArray, 0);


        System.out.println("Extracted Substring: " + new String(targetArray));
    }

    public static void main(String[] args) {

        College myCollege = new College("Welcome to Bmsce college");


        System.out.println(myCollege);


        myCollege.extractSubstring();
    }
}
```

OUTPUT:

```
C:\Users\Admin\Desktop>javac College.java

C:\Users\Admin\Desktop>java College
College{name='Welcome to Bmsce college'}
Extracted Substring: Bmsce
```

SOURCE CODE:

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

      String str1 = "Bmsce";
      String str2 = "Bmsce";
      boolean isEqual = str1.equals(str2);
      System.out.println(str1 + " equals " + str2 + " -> " + isEqual);

      String str3 = "Bmsce";
      String str4 = "College";
      isEqual = str3.equals(str4);
      System.out.println(str3 + " equals " + str4 + " -> " + isEqual);


      String str5 = "Bmsce";
      String str6 = "BMSCE";
      isEqual = str5.equals(str6);
      System.out.println(str5 + " equals " + str6 + " -> " + isEqual);


      String str7 = "Bmsce";
      String str8 = "BMSCE";
      boolean isEqualIgnoreCase = str7.equalsIgnoreCase(str8);
      System.out.println(str7 + " equalsIgnoreCase " + str8 + " -> " + isEqualIgnoreCase);
   }
}
```

OUTPUT:

```
C:\Users\Admin\Desktop>javac StringConcatenation.java

C:\Users\Admin\Desktop>java StringConcatenation
Concatenated String: helloworld

C:\Users\Admin\Desktop>javac StringComparison.java

C:\Users\Admin\Desktop>java StringComparison
Bmsce equals Bmsce -> true
Bmsce equals College -> false
Bmsce equals BMSCE -> false
Bmsce equalsIgnoreCase BMSCE -> true
```

```
public class Find{
public static void main(String args[]){
String str1="Welcome to BMSCE College of Engineering";
 String otherstr="BMSCE College";
Boolean ismatch=str1.regionMatches(true, 11,otherstr,0,otherstr.length());
if(ismatch)
 System.out.println("substring is matched");
else
System.out.println("substring is not matched");
}
}
```

OUTPUT:

```
C:\Users\Admin\Desktop>javac Find.java

C:\Users\Admin\Desktop>java Find
substring is matched
```

PROGRAM-8 and 9

SOURCE CODE:

```
public class StartEndDemo {
    public static void main(String[] args) {
        String mainString1 = "Hello, World!";
        String pre1 = "Hello";
        boolean startsWith1 = mainString1.startsWith(pre1);
        System.out.println( mainString1 +"starts with"+ pre1 + " -> " + startsWith1);

        String mainString2 = "Java Programming";
        String prefix2 = "Python";
        boolean startsWith2 = mainString2.startsWith(prefix2);
        System.out.println( mainString2 + " starts with " + prefix2 + " -> " + startsWith2);

        String mainString3 = "Hello, World!";
        String suffix1 = "World!";
        boolean endsWith1 = mainString3.endsWith(suffix1);
        System.out.println( mainString3 + "ends with " + suffix1 + " -> " + endsWith1);

        String mainString4 = "Java Programming";
        String suffix2 = "C++";
        boolean endsWith2 = mainString4.endsWith(suffix2);
        System.out.println(mainString4 + " ends with " + suffix2 + " -> " + endsWith2);
    }
}
```

OUTPUT:

```
C:\Users\Admin\Desktop>javac StartEndDemo.java

C:\Users\Admin\Desktop>java StartEndDemo
Hello, World!starts withHello -> true
Java Programming starts with Python -> false
Hello, World!ends with World! -> true
Java Programming ends with C++ -> false
```

PROGRAM-10

SOURCE CODE:

```
class EqualsNotEqualTo{
public static void main(String args[]){
String s1="Hello!";
String s2= new String(s1);
System.out.println(s1 + " equals " + s2 + " -> " + s1.equals(s2));
System.out.println(s1 + " == " + s2 + " -> " + (s1 == s2));
  }
}
```

```
C:\Users\Admin\Desktop>javac EqualsNotEqualTo.java

C:\Users\Admin\Desktop>java EqualsNotEqualTo
Hello! equals Hello! -> true
Hello! == Hello! -> false
```

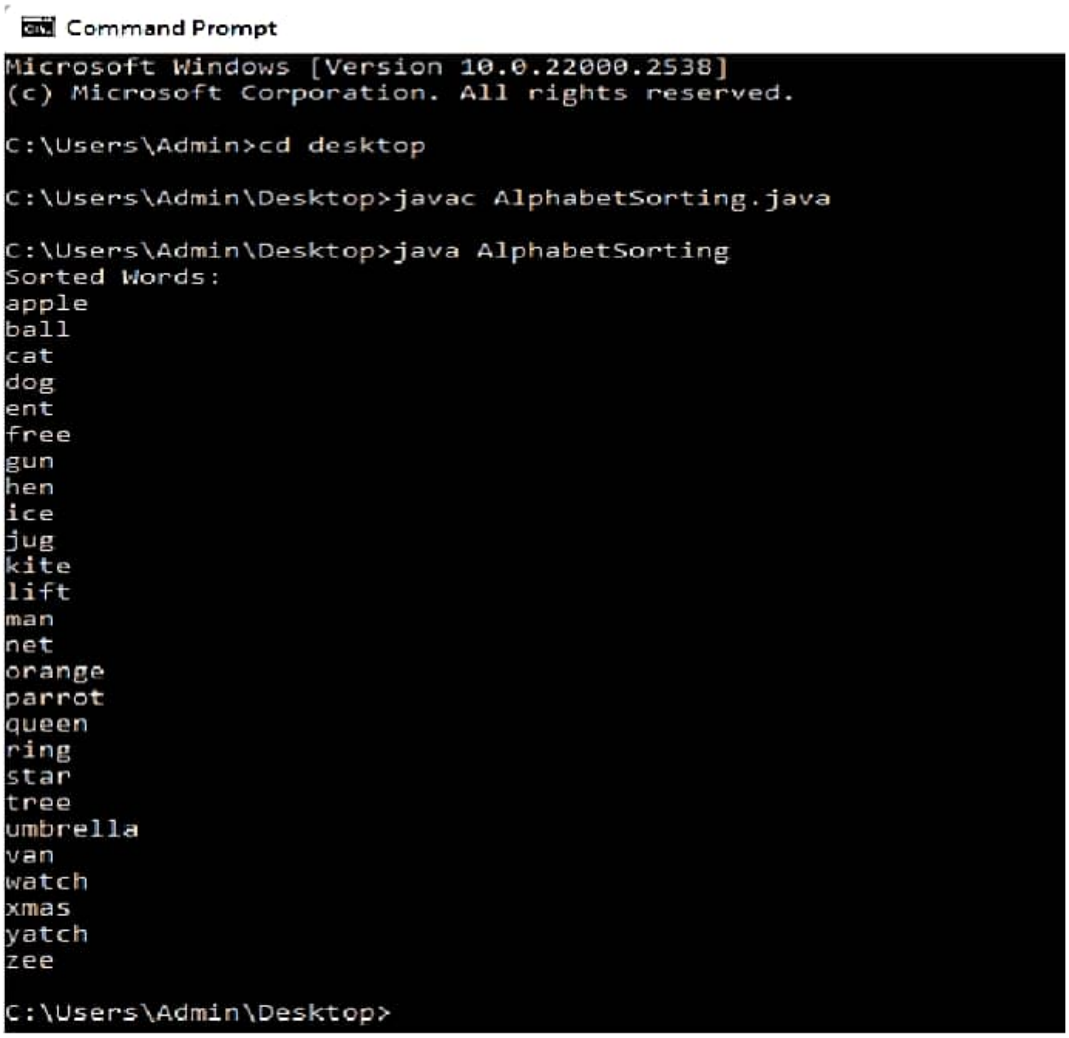
PROGRAM-11

SOURCE CODE:

```
import java.util.Arrays;

public class AlphabetSorting {
   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);
      }
```

```
    }
}
```

OUTPUT:



PROGRAM-12

SOURCE CODE:

```
import java.util.Arrays;

public class NumberSorting {
    public static void main(String[] args) {
        Integer[] numbers = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};

        Arrays.sort(numbers, (num1, num2) -> num2.compareTo(num1));

        System.out.println("Sorted Numbers (Descending Order):");
        for (Integer number : numbers) {
            System.out.println(number);
        }
    }
}
```

OUTPUT:

```
C:\Users\Admin\Desktop>javac NumberSorting.java

C:\Users\Admin\Desktop>java NumberSorting
Sorted Numbers (Descending Order):
10
9
8
7
6
5
4
3
2
1
```

PROGRAM-13

SOURCE CODE:

```
public class StringReplacement {
   public static void main(String[] args) {
      String originalString = "Thwas was a test. Thwas was, too.";
      int indexOfWas = originalString.indexOf("was");
      while (indexOfWas != -1) {

         String updatedString = originalString.substring(0, indexOfWas) + "is" +
originalString.substring(indexOfWas + "was".length());

         originalString = updatedString;

         indexOfWas = originalString.indexOf("was");
      }

      System.out.println("Modified String: " + originalString);
   }
}
```

OUTPUT:

```
C:\Users\Admin\Desktop>javac StringReplacement.java

C:\Users\Admin\Desktop>java StringReplacement
Modified String: This is a test. This is, too.
```

PROGRAM-14

SOURCE CODE:

```
public class StringConcatenation {
```

```
    public static void main(String[] args) {
        String s1 = "hello";
        String s2 = "world";

        String result = s1.concat(s2);

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

OUTPUT:

```
C:\Users\Admin\Desktop>javac StringConcatenation.java

C:\Users\Admin\Desktop>java StringConcatenation
Concatenated String: helloworld
```

PROGRAM-15

SOURCE CODE:

```
 public class StringReplaceDemo {
public static void main(String[] args) {
String originalString = &amp;quot;This is my College.&amp;quot;;
String modifiedString = originalString.replace(&amp;quot;College&amp;quot;, &amp;quot;Commege&amp;quot;);
System.out.println(&amp;quot;Original String: &amp;quot; + originalString);
System.out.println(&amp;quot;Modified String: &amp;quot; + modifiedString);

}
}
```

OUTPUT:

```
C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac StringReplaceDemo.java

C:\Users\Admin\Desktop>java StringReplaceDemo
Original String: This is my College.
Modified String: This is my Commege.
```

PROGRAM-16

SOURCE CODE:

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

String originalString = &amp;quot; Hello Friends &amp;quot;;

String trimmedString = originalString.trim();

System.out.println(&amp;quot;Original String: &amp;#39;&amp;quot; + originalString + &amp;quot;&amp;#39;&amp;quot;);
```

System.out.println("Trimmed String: '" + trimmedString + "'");
}
}

OUTPUT:

```
C:\Users\Admin\Desktop> javac StringTrimDemo.java

C:\Users\Admin\Desktop>java StringTrimDemo
Original String: '  Hello Friends  '
Trimmed String: 'Hello Friends'
```

PROGRAM-17

SOURCE CODE:

```
import java.util.Arrays;
import java.util.Scanner;

class Student {
private int regNumber;
private String fullName;
private short semester;
private float cgpa;

public Student() {
this.regNumber = 0;
this.fullName = &amp;quot;&amp;quot;;
this.semester = 0;
this.cgpa = 0.0f;
}

public Student(int regNumber, String fullName, short semester, float cgpa) {
this.regNumber = regNumber;
this.fullName = fullName;
this.semester = semester;
this.cgpa = cgpa;
}

public void display() {
System.out.println(&amp;quot;Registration Number: &amp;quot; + regNumber);
System.out.println(&amp;quot;Full Name: &amp;quot; + fullName);
System.out.println(&amp;quot;Semester: &amp;quot; + semester);
System.out.println(&amp;quot;CGPA: &amp;quot; + cgpa);
System.out.println();
}

public float getCGPA() {
return cgpa;
}
```

System.out.println("Trimmed String: '" + trimmedString + "'");
}
}

OUTPUT:

```
C:\Users\Admin\Desktop> javac StringTrimDemo.java

C:\Users\Admin\Desktop>java StringTrimDemo
Original String: ' Hello Friends '
Trimmed String: 'Hello Friends'
```

PROGRAM-17

SOURCE CODE:

```
import java.util.Arrays;
import java.util.Scanner;

class Student {
private int regNumber;
private String fullName;
private short semester;
private float cgpa;

public Student() {
this.regNumber = 0;
this.fullName = &amp;quot;&amp;quot;;
this.semester = 0;
this.cgpa = 0.0f;
}

public Student(int regNumber, String fullName, short semester, float cgpa) {
this.regNumber = regNumber;
this.fullName = fullName;
this.semester = semester;
this.cgpa = cgpa;
}

public void display() {
System.out.println(&amp;quot;Registration Number: &amp;quot; + regNumber);
System.out.println(&amp;quot;Full Name: &amp;quot; + fullName);
System.out.println(&amp;quot;Semester: &amp;quot; + semester);
System.out.println(&amp;quot;CGPA: &amp;quot; + cgpa);
System.out.println();
}

public float getCGPA() {
return cgpa;
}
```

```
public String getFullName() {
return fullName;
}
}

public class StudentRecords {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);

Student[] students = new Student[5];

for (int i = 0; i &lt; students.length; i++) {
System.out.println(&quot;Enter details for Student &quot; + (i + 1) + &quot;:&quot;);
System.out.print(&quot;Registration Number: &quot;);
int regNumber = scanner.nextInt();
scanner.nextLine(); // Consume the newline
System.out.print(&quot;Full Name: &quot;);
String fullName = scanner.nextLine();
System.out.print(&quot;Semester: &quot;);
short semester = scanner.nextShort();
System.out.print(&quot;CGPA: &quot;);
float cgpa = scanner.nextFloat();

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

System.out.println(&quot;Displaying Student Records:&quot;);
for (Student student : students) {
student.display();
}

Arrays.sort(students, (s1, s2) -&gt; Float.compare(s2.getCGPA(), s1.getCGPA()));

System.out.println(&quot;Student Records Sorted by CGPA:&quot;);
for (Student student : students) {
student.display();

}

Arrays.sort(students, (s1, s2) -&gt; s1.getFullName().compareTo(s2.getFullName()));

System.out.println(&quot;Student Records Sorted by Name:&quot;);
for (Student student : students) {
student.display();
}
```

OUTPUT:

```
C:\Users\Admin\Desktop>javac Studer

C:\Users\Admin\Desktop>java Student
Enter details for Student 1:
Registration Number: 1
Full Name: A
Semester: 1
CGPA: 9
Enter details for Student 2:
Registration Number: 2
Full Name: B
Semester: 1
CGPA: 8.1
Enter details for Student 3:
Registration Number: 3
Full Name: C
Semester: 1
CGPA: 9.3
Enter details for Student 4:
Registration Number: 4
Full Name: d
Semester: 1
CGPA: 8.7
Enter details for Student 5:
Registration Number: 5
Full Name: e
Semester: 1
CGPA: 9.6
Displaying Student Records:
Registration Number: 1
Full Name: A
Semester: 1
CGPA: 9.0

Registration Number: 2
Full Name: B
Semester: 1
CGPA: 8.1

Registration Number: 3
Full Name: C
Semester: 1
CGPA: 9.3

Registration Number: 4
Full Name: d
Semester: 1
CGPA: 8.7

Registration Number: 5
Full Name: e
Semester: 1
CGPA: 9.6

Student Records Sorted by CGPA:
Registration Number: 5
Full Name: e
Semester: 1
CGPA: 9.6

Registration Number: 3
Full Name: C
Semester: 1
```

```
Registration Number: 3
Full Name: C
Semester: 1
CGPA: 9.3

Registration Number: 1
Full Name: A
Semester: 1
CGPA: 9.0

Registration Number: 4
Full Name: d
Semester: 1
CGPA: 8.7

Registration Number: 2
Full Name: B
Semester: 1
CGPA: 8.1

Student Records Sorted by Name:
Registration Number: 1
Full Name: A
Semester: 1
CGPA: 9.0

Registration Number: 2
Full Name: B
Semester: 1
CGPA: 8.1

Registration Number: 3
Full Name: C
Semester: 1
CGPA: 9.3

Registration Number: 4
Full Name: d
Semester: 1
CGPA: 8.7

Registration Number: 5
Full Name: e
Semester: 1
CGPA: 9.6
```

PROGRAM-18

SOURCE CODE:

public class

```
StringBufferDemo {
public static void main(String[] args) {
StringBuffer stringBuffer = new StringBuffer(&amp;quot;Hello, StringBuffer!&amp;quot;);

stringBuffer.setLength(5);
System.out.println(&amp;quot;After setLength(5): &amp;quot; + stringBuffer);

char charAtIndex = stringBuffer.charAt(1);
System.out.println(&amp;quot;Character at index 1: &amp;quot; + charAtIndex);

stringBuffer.setCharAt(1, &amp;#39;a&amp;#39;);
System.out.println(&amp;quot;After setCharAt(1, &amp;#39;a&amp;#39;): &amp;quot; + stringBuffer);

char[] charArray = new char[5];
stringBuffer.getChars(0, 5, charArray, 0);
System.out.println(&amp;quot;Characters from index 0 to 4: &amp;quot; + new String(charArray));
stringBuffer.append(&amp;quot; Appended!&amp;quot;);
System.out.println(&amp;quot;After append(): &amp;quot; + stringBuffer);

stringBuffer.insert(7, &amp;quot;Inserted &amp;quot;);
System.out.println(&amp;quot;After insert(7, &amp;#39;Inserted &amp;#39;): &amp;quot; + stringBuffer);

stringBuffer.reverse();
System.out.println(&amp;quot;After reverse(): &amp;quot; + stringBuffer);

stringBuffer.delete(5, 14);
System.out.println(&amp;quot;After delete(5, 14): &amp;quot; + stringBuffer);

stringBuffer.deleteCharAt(0);
System.out.println(&amp;quot;After deleteCharAt(0): &amp;quot; + stringBuffer);
stringBuffer.replace(0, 4, &amp;quot;Replaced&amp;quot;);
System.out.println(&amp;quot;After replace(0, 4, &amp;#39;Replaced&amp;#39;): &amp;quot; + stringBuffer);

String substring = stringBuffer.substring(3, 8);
System.out.println(&amp;quot;Substring from index 3 to 7: &amp;quot; + substring);
}

}
```

OUTPUT:

```
C:\Users\Admin\Desktop>javac StringBufferDemo.java

C:\Users\Admin\Desktop>java StringBufferDemo
After setLength(5): Hello
Character at index 1: e
After setCharAt(1, 'a'): Hallo
Characters from index 0 to 4: Hallo
After append(): Hallo Appended!
After insert(7, 'Inserted '): Hallo AInserted ppended!
After reverse(): !dednepp detresnIA ollaH
After delete(5, 14): !dednsnIA ollaH
After deleteCharAt(0): dednsnIA ollaH
After replace(0, 4, 'Replaced'): ReplacedsnIA ollaH
Substring from index 3 to 7: laced
```

PROGRAM-19

SOURCE CODE:

```
Abstract class Bird
abstract class Bird {
abstract void fly();
abstract void makeSound();
}

class Eagle extends Bird {
void fly() {
System.out.println(&quot;Eagle flies high in the sky with powerful wings.&quot;);
}

void makeSound() {
System.out.println(&quot;Eagle makes a sharp and distinctive cry.&quot;);
}
}

class Hawk extends Bird {
void fly() {
System.out.println(&quot;Hawk soars through the air with agile maneuvers.&quot;);
}

void makeSound() {
System.out.println(&quot;Hawk emits a high-pitched screech while flying.&quot;);
}
}
public class BirdTest {
public static void main(String[] args) {
```

```
Eagle eagle = new Eagle();
Hawk hawk = new Hawk();

System.out.println(&amp;quot;Details about Eagle:&amp;quot;);
eagle.fly();
eagle.makeSound();

System.out.println(&amp;quot;\nDetails about Hawk:&amp;quot;);
hawk.fly();
hawk.makeSound();
}
}
```

OUTPUT:

```
C:\Users\Admin\Desktop>javac BirdTest.java

C:\Users\Admin\Desktop>java BirdTest
Details about Eagle:
Eagle flies high in the sky with powerful wings.
Eagle makes a sharp and distinctive cry.

Details about Hawk:
Hawk soars through the air with agile maneuvers.
Hawk emits a high-pitched screech while flying.
```

PROGRAM-20

SOURCE CODE:

```
abstract class Shape {
abstract double calculateArea();
abstract double calculatePerimeter();
}

class Circle extends Shape {

private double radius;

public Circle(double radius) {
this.radius = radius;
}

double calculateArea() {
return Math.PI * radius * radius;
}

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;
}

double calculateArea() {
double s = (side1 + side2 + side3) / 2.0;
return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));
}

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(&quot;Details about Circle:&quot;);
System.out.println(&quot;Area: &quot; + circle.calculateArea());
System.out.println(&quot;Perimeter: &quot; + circle.calculatePerimeter());

System.out.println(&quot;\nDetails about Triangle:&quot;);
System.out.println(&quot;Area: &quot; + triangle.calculateArea());
System.out.println(&quot;Perimeter: &quot; + triangle.calculatePerimeter());
}
}
```

OUTPUT:

```
:\Users\Admin\Desktop>javac ShapeTest.java

:\Users\Admin\Desktop>java ShapeTest
etails about Circle:
rea: 78.53981633974483
erimeter: 31.41592653589793

etails about Triangle:
rea: 6.0
erimeter: 12.0
```