

INDEX

Sr no.	<u>Assignment 1</u>	Pg no.	signature
Q.1	Write a java program print “Hello World”.		
Q.2	Write a java program to show uses of final keyword with variable, method and class in Java.		
Q.3	Write a java program to display uses of abstract keyword in Java.		
Q.4	Write a java program to illustrate implementing an interface and extending a class in Java.		
Q.5	Write a java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula.		
Q.6	The following rule defines the Fibonacci sequence. The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a java program that uses both recursive and non-recursive functions		
	<u>Assignment 2</u>		
Q.1	Write a java program to multiply two given matrices.		
Q.2	Write a java program to implement a) Method overloading and constructors overloading b) Write a java program to implement method overriding.		
Q.3	Write a java program to check whether a given string is a palindrome.		
Q.4	Write a java program for sorting a given list of names in ascending order		
Q.5	Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given Shape		
Q.6	Write a Java program to show uses of Vector class.		

Sr no.	<u>Assignment 3</u>	Pg no.	signature
Q.1	Write a Java program to write a string in a file using FileOutputStream class.		
Q.2	Write a Java program to read content of a file and print it on screen using FileInputStream class.		
Q.3	Write a Java program which takes input from a file and write it into another file.		
Q.4	Write a Java program to show uses of try-catch in exception handling.		
Q.5	Write a Java program to demonstrate throwing user defined exceptions as well as finally in exception handling.		
Q.6	Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box		
	<u>Assignment 4</u>		
Q.1	Write a java program that implements a multi-thread application that has three threads. First thread generates a random integer every 1 second, and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.		
Q.2	Write a java program that displays the number of characters, lines, and words in a text file.		
Q.3	Write a java program that reads a file and displays the file on the screen with a line number before each line.		
Q.4	Write a java program that loads names and phone numbers from a text file where the data is organized as one line per record, and each field in a record is separated by a tab (/t). It takes a name or phone number as input and prints the corresponding other value from the hash table. Hint: Use hash tables		
Q.5	Write a java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.		

ASSIGNMENT-1

Q1. Write a java program print "Hello World".

Program:

```
public class Hello{
    public static void main(String args[])
    {
        System.out.println("Hello World");
    }
}
```

Output:

Hello World

Q2. Write a java program to show uses of final keyword with variable, method and class in Java.

Program:

```
// final class
final class motor{
    // final variable
    final String model = "Mahindra";
    // final method
    public final void start()
    {System.out.println("Vehicle Started.");}
}
class Bike {
    public final void start()
    {
        System.out.println("Bike Started.");
    }
}
public class FinalVariable {
    public static void main(String[] args) {
        motor mo = new motor();
        System.out.println("Model : " +mo.model);
        mo.start();
        Bike bike = new Bike();
        bike.start();
    }
}
```

Output:

Model : Mahindra
Vehicle Started.
Bike Started.

Q3. Write a java program to display uses of abstract keyword in Java.

Program:

```
abstract class Motorcar{
    String msg;
    Motorcar(String msg)
    {
        this.msg = msg;
    }
    void display()
    {
        System.out.println(msg);
    }
}
class HondaGadi extends Motorcar{
    HondaGadi(String msg)
    {
        super(msg);
    }
}
public class AbstractKeyword {
    public static void main(String[] args) {
        HondaGadi ob = new HondaGadi("This is Example of Abstract Keyword.");
        ob.display();
    }
}
```

Output:

This is Example of Abstract Keyword.

Q4. Write a java program to illustrate implementing an interface and extending a class in Java.

Program:

```
//define an interface
interface Interface {
    void draw();
}
class Shape {
    void describe(){
        System.out.println("This is a shape");
    }
}
// Define a class Circle that extends shape and implement Interface class
class Circle extends Shape implements Interface {
    @Override
    public void draw(){
        System.out.println("Drawing a circle.");
    }
}
public class Main{
    public static void main(String args[]){
        Circle circle = new Circle();
        circle.describe();
        circle.draw();
    }
}
```

Output:

This is a shape
Drawing a circle.

Q5. Write a java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula.

Program:

```
import java.util.*;
class Roots {
    public static void main(String args[]){
        double r1,r2; //ax^2+bx+c=0
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the coefficients a,b and c: ");
        System.out.print("a: ");double a = input.nextDouble();
        System.out.print("b: ");double b = input.nextDouble();
        System.out.print("c: ");
        double c = input.nextDouble();
        //d = discriminant
        double d = (b*b) - (4*a*c);
        if(d==0) //real and equal
        { System.out.println("Roots are Real and Equal.");
          r1=r2= -b/(2*a);
          System.out.println("R1= " +r1);
          System.out.println("R2= " +r2);}
        else if(d>0) // real and distinct
        { System.out.println("Roots are real and distinct.");
          r1 = (-b + Math.sqrt(d))/(2*a);
          r2 = (-b - Math.sqrt(d))/(2*a);
          System.out.println("R1= " +r1);
          System.out.println("R2= " +r2);}
        else //distinct and imaginary
        { System.out.println("Roots are distinct and imaginary.");
          double x = -b/(2*a);
          double y = Math.sqrt(-d)/(2*a);
          System.out.println("R1= " +x + "+i" + y);
          System.out.println("R2= " +x + "-i" + y);}}}
```

Output:

```
Enter the coefficients a,b and c:
a: 2
b: 3
c: 4
Roots are distinct and imaginary.
R1= -0.75+i1.1989578808281798
R2= -0.75-i1.1989578808281798
```

Q6.The following rule defines the Fibonacci sequence. The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a java program that uses both recursive and non-recursive functions.

Program:

```
import java.util.*;
class Fabonacci {
    public static int fibonacciRecursive(int n){
        if(n<=1){return n;}
        else{
            return fibonacciRecursive(n-1) + fibonacciRecursive(n-2);
        }
    }
    public static void fibonacciNonRecursive(int n){
        int[] fibonacciArray = new int[n];
        fibonacciArray[0] =1;
        fibonacciArray[1] =1;
        System.out.println("Fibonacci Sequence for Non-Recursive Method: ");
        for(int i=2;i<n;i++){
            fibonacciArray[i] = fibonacciArray[i-1] + fibonacciArray[i-2];}
        for(int i=0; i<n;i++){
            System.out.println(fibonacciArray[i] + " ");}
        System.out.println();}
    public static void main(String args[]){
        Scanner num = new Scanner(System.in);
        System.out.print("Enter value of n: ");
        int n = num.nextInt();
        System.out.println("Fibonacci Sequence for Recursive Method : ");
        for(int i=0;i<n;i++){
            System.out.println(fibonacciRecursive(i) + " "); }
        System.out.println();
        fibonacciNonRecursive(n);
    }
}
```

Output:

```
Enter value of n: 10
Fibonacci Sequence for Recursive Method :
1 1 2 3 5 8 13 21 34 55
Fibonacci Sequence for Non-Recursive Method:
1 1 2 3 5 8 13 21 34 55
```

ASSIGNMENT-2

Q1. Write a java program to multiply two given matrices.

Program:

```
public class MultiplyTwoMatrix{
    public static void main(String args[]){
        int a[][] = {
            {1,2,3},
            {4,5,6},
            {13,14,15}
        };
        int b[][]={
            {7,8,9},
            {10,11,12},
            {16,17,18}
        };
        int c[][] = new int [3][3]; // to store multiply of matrix
        System.out.println("Multiplications of two matrix -> ");
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                c[i][j]=0;
                for(int k=0;k<3;k++)
                {
                    c[i][j]=a[i][k]*b[k][j];
                }
                System.out.print(c[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
Multiplications of two matrix ->
48  51  54
96  102 108
240 255 270
```


Q2. Write a java program to implement:-

- (a) Method overloading and constructors overloading.
- (b) Write a java program to implement method overriding.

Answer:

(a) Method overloading and constructors overloading.

Program:

```
public class perimeter{
    int length;
    int breadth;
    //default constructor
    perimeter()
    {
        length=0;
        breadth=0;
    }
    //parameterized constructor
    perimeter(int x,int y)
    {
        length = x;
        breadth = y;
    }
    void cal_perimeter()
    {
        int peri;
        peri = 2*(length + breadth);
        System.out.println("\n The parameter of Rectangel is: " + peri);
    }
    public static void main(String args[])
    {
        perimeter p1 = new perimeter();
        perimeter p2 = new perimeter(5,10);
        p1.cal_perimeter();
        p2.cal_perimeter();
    }
}
```

Output:

The parameter of Rectangel is: 0
The parameter of Rectangel is: 30

(b) Write a java program to implement method overriding.

Program:

```
class Vehicle {
    void run()
    {
        System.out.println("Vehicle is running.");
    }
}

class Bike extends Vehicle{
    @Override
    void run(){
        System.out.println("Bike is not safe to Travel.");
    }
}

class MethodOverriding{
    public static void main(String args[])
    {
        Bike ob1 = new Bike();
        ob1.run();
        Vehicle ob2 = new Bike();
        ob2.run();
        Vehicle ob3 = new Vehicle();
        ob3.run();
    }
}
```

Output:

```
Bike is not safe to Travel.
Bike is not safe to Travel.
Vehicle is running.
```

Q3. Write a java program to check whether a given string is a palindrome.

Program:

```
import java.util.*;

public class PalindromeChecker{
    public static boolean isPalindrome(String str){
        str = str.replaceAll("\\s", "").toLowerCase();
        int left=0;
        int right=str.length()-1;
        while(left<right){
            if(str.charAt(left) != str.charAt(right)){
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);
        System.out.print("Enter string to check it is Palindrome or not: ");
        String inputstr = input.nextLine();
        if(isPalindrome(inputstr)){
            System.out.println("The string is a palindrome.");
        }
        else{
            System.out.println("The string is not a palindrome.");
        }
    }
}
```

Output:

```
Enter string to check it is Palindrome or not: abba
The string is a palindrome
```

Q4. Write a java program for sorting a given list of names in ascending order.

Program:

```
import java.util.Arrays;

public class SortingString {
    public static void main(String[] args) {
        String str = "I am sumit Khowal From MCA";
        // convert the string to character array
        char[] charArray = str.toCharArray();
        // to sort
        Arrays.sort(charArray);
        String sortedstr = new String (charArray);
        System.out.println("Original String : \n"+str);
        System.out.println(" Sorted String : \n"+ sortedstr);
    }
}
```

Output:

```
Original String :
I am sumit Khowal From MCA
Sorted String :
ACFIKMaahilmmmoorstuw
```

Q5. Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given Shape.

Program:

```
abstract class Shape{
    protected int length;
    protected int width;
    public Shape(int length,int width){
        this.length = length;
        this.width = width;
    }
    public abstract void printArea();
}

class Rectangle extends Shape{
    public Rectangle(int length,int width){
        super(length,width);}
}
```

```

@Override
public void printArea(){
    int area=length*width;
    System.out.println("Rectangle Area: " + area);
}
}
class Triangle extends Shape{
    public Triangle(int length,int width){
        super(length,width);
    }
    @Override
    public void printArea(){
        double area = 0.5 *length * width;
        System.out.println("Triangle Area: " + area);
    }
}
class Circle extends Shape{
    public Circle(int radius){
        super(radius,radius);
    }
    @Override
    public void printArea(){
        double area = Math.PI * length * length;
        String formattedArea = String.format("%.2f", area);
        System.out.println("Circle Area: "+ formattedArea);
    }
}
public class ShapeTest {
    public static void main(String args[]){
        Shape rectangle = new Rectangle(5,4);
        Shape triangle = new Triangle(6,3);
        Shape circle = new Circle(5);
        System.out.println("The calculated Areas of Rectangle,Triangle and Circle : ");
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}

```

Output:

The calculated Areas of Rectangle,Triangle and Circle :
 Rectangle Area: 20
 Triangle Area: 9.0
 Circle Area: 78.54

Q6. Write a Java program to show uses of Vector class.

Program:

```
import java.util.Vector;

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

        //create a vector of string
        Vector<String> vector = new Vector<>();
        vector.add("Apple ");
        vector.add("Banana ");
        vector.add("Cherry ");
        vector.add("Orange ");
        vector.add("Watermelon");
        System.out.println("Elements in the Vector: ");
        for (String fruit: vector){
            System.out.println(fruit);
        }
    }
}
```

Output:

```
Elements in the Vector:
Apple
Banana
Cherry
Orange
Watermelon
```

ASSIGNMENT-3

Q1. Write a Java program to write a string in a file using FileOutputStream class.

Program:

```
import java.io.FileOutputStream;
import java.io.IOException;

public class FileStringToFile
{
    public static void main(String[] args)
    {
        String content = "Hello, Sumit is writing to the file by FileOutputStream.";

        // Specify the file path
        String filePath = "output.txt";

        try (FileOutputStream fos = new FileOutputStream(filePath)) {
            // Convert the string to bytes and write it to the file
            byte[] bytes = content.getBytes();
            fos.write(bytes);

            System.out.println("String has been written to the file.");
        } catch (IOException e) {
            System.err.println("An error occurred: " + e.getMessage());
        }
    }
}
```

Output:

String has been written to the file

Q2. Write a Java program to read content of a file and print it on screen using FileInputStream class.

Program:

```
import java.io.FileInputStream;
import java.io.IOException;

public class ReadFileContent {
    public static void main(String args[] ){
        String filePath = "input.txt";
        try (FileInputStream fis = new FileInputStream(filePath)) {
            int byteRead;
            StringBuilder contentBuilder = new StringBuilder();
            while ((byteRead = fis.read()) != -1) {
                contentBuilder.append((char) byteRead);
            }
            System.out.println(contentBuilder.toString());
        } catch (IOException e){
            System.out.println("An error occurred: " + e.getMessage());
        }
    }
}
```

Output:

This is input file content by using FileInputStream.

Q3. Write a Java program which takes input from a file and write it into another file.

Program:

```
import java.io.*;

public class CopyFileToAnother {
    public static void main(String[] args) {
        // Specify the source file path
        String sourceFilePath = "input.txt"; // Change this to the path of your source file

        // Specify the destination file path
        String destinationFilePath = "destination.txt"; // Change this to the path of your destination file

        try (FileReader fileReader = new FileReader(sourceFilePath);
            BufferedReader bufferedReader = new BufferedReader(fileReader);
            FileWriter fileWriter = new FileWriter(destinationFilePath);
            BufferedWriter bufferedWriter = new BufferedWriter(fileWriter)) {

            String line;
```



```

        while ((line = bufferedReader.readLine()) != null) {
            bufferedWriter.write(line);
            bufferedWriter.newLine(); // Add a newline character to separate lines in
the destination file
        }

        System.out.println("File copied successfully.");
    } catch (IOException e) {
        System.err.println("An error occurred: " + e.getMessage());
    }
}
}

```

Output:

File copied successfully.

Q4. Write a Java program to show uses of try-catch in exception handling.

Program:

```

public class ExceptionHandling
{
    public static void main(String[] args)
    {
        try {
            int[] numbers = {1, 2, 3};
            int result = numbers[5] / 0;
            System.out.println("Result: " + result);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("An ArrayIndexOutOfBoundsException occurred: " +
e.getMessage());
        } catch (ArithmeticException e) {
            System.out.println("An ArithmeticException occurred: " + e.getMessage());
        }
        System.out.println("Program continues after exception handling.");
    }
}

```

Output:

Program continues after exception handling.

Q5. Write a Java program to demonstrate throwing user defined exceptions as well as finally in exception handling.

Program:

```
// Define a custom exception class
class NegativeNumberException extends Exception
{
    public NegativeNumberException(String message)
    {
        super(message);
    }
}

public class CustomException{
    public static void main(String[] args) {
        try {
            int number = -5;
            if (number < 0) {
                throw new NegativeNumberException("Number cannot be negative");
            }
            System.out.println("Number: " + number);
        } catch (NegativeNumberException e) {
            System.out.println("Custom exception caught: " + e.getMessage());
        }

        System.out.println("Program continues after exception handling.");
    }
}
```

Output:

Custom exception caught: Number cannot be negative
Program continues after exception handling.

Q6. Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Program:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class DivisionCalculator {
    private JFrame frame;
    private JPanel panel;
    private JTextField num1Field;
    private JTextField num2Field;
    private JButton divideButton;
    private JLabel resultLabel;

    public DivisionCalculator() {
        // Create the JFrame and JPanel
        frame = new JFrame("Integer Division Calculator");
        panel = new JPanel();

        // Create input fields, button, and result label
        num1Field = new JTextField(10);
        num2Field = new JTextField(10);
        divideButton = new JButton("Divide");
        resultLabel = new JLabel("Result:");

        // Add components to the panel
        panel.add(new JLabel("Num1:"));
        panel.add(num1Field);
        panel.add(new JLabel("Num2:"));
        panel.add(num2Field);
        panel.add(divideButton);
        panel.add(resultLabel);
    }
}
```

```

// Add action listener to the Divide button
divideButton.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try {
            int num1 = Integer.parseInt(num1Field.getText());
            int num2 = Integer.parseInt(num2Field.getText());

            // Handle division by zero
            if (num2 == 0)
            {
                throw new ArithmeticException("Division by zero");
            }

            int result = num1 / num2;
            resultLabel.setText("Result: " + result);

        } catch (NumberFormatException ex)
        {
            // Handle NumberFormatException
            JOptionPane.showMessageDialog(frame, "Invalid input. Please enter
valid integers.", "Error", JOptionPane.ERROR_MESSAGE);
        } catch (ArithmeticException ex)
        {
            // Handle ArithmeticException (division by zero)
            JOptionPane.showMessageDialog(frame, "Division by zero is not
allowed.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});

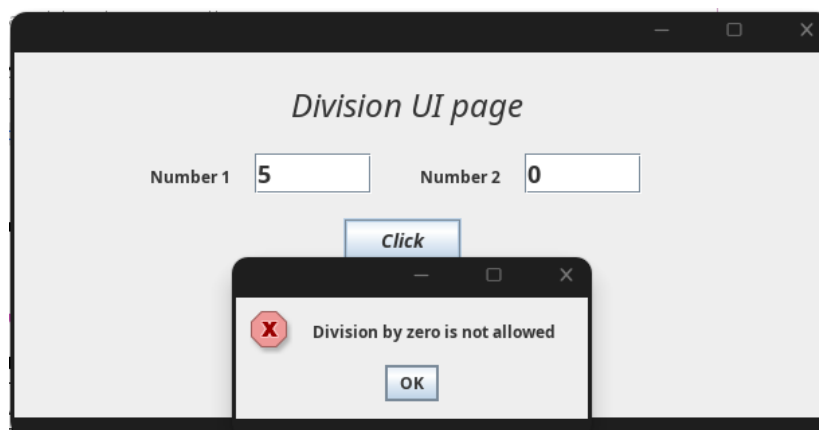
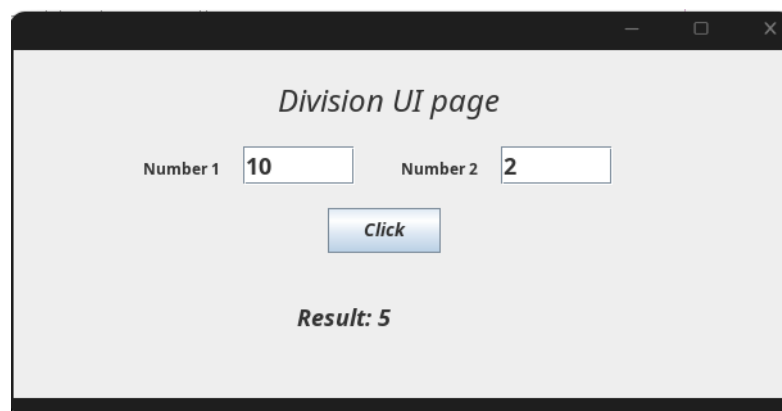
// Set layout and add panel to the frame
frame.setLayout(new BorderLayout());
frame.add(panel, BorderLayout.CENTER);

// Set frame properties
frame.setSize(300, 150);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
}

```

```
public static void main(String[] args) {  
    SwingUtilities.invokeLater(new Runnable() {  
        public void run() {  
            new DivisionCalculator();  
        }  
    });  
}
```

Output:



(This is Arithmetic Error)

ASSIGNMENT-4

Q1. Write a java program that implements a multi- thread application that has three threads. First thread generates a random integer every 1 second, and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.

Program:

```
import java.util.Random;
```

```
class RandomNumberGenerator extends Thread
{
    @Override
    public void run(){
        Random random = new Random();
        while(true){
            try{
                Thread.sleep(1000); //sleep for 1 second;
                int randomNumber = random.nextInt(100);
                if(randomNumber % 2 == 0){
                    SquareThread squareThread = new SquareThread(randomNumber);
                    squareThread.start();
                }
                else{
                    CubeThread cubeThread = new CubeThread(randomNumber);
                    cubeThread.start();
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class SquareThread extends Thread{
    private int number;
    public SquareThread(int number){
        this.number = number;
    }
    @Override
    public void run(){
```

```

    int square = number * number;
    System.out.println("Square of " + number + " is : " + square);
}
}
class CubeThread extends Thread{
    private int number;
    public CubeThread(int number){
        this.number= number;
    }
    @Override
    public void run(){
        int cube = number * number * number;
        System.out.println("Cube of " + number + " is: " + cube);
    }
}
public class MultiThread
{
    public static void main(String args[])
    {
        RandomNumberGenerator randomNumberGenerator = new
RandomNumberGenerator();
        randomNumberGenerator.start();
    }
}

```

Output:

Square of 68 is : 4624
 Square of 2 is : 4
 Square of 14 is : 196
 Cube of 65 is: 274625
 Cube of 11 is: 1331
 Square of 10 is : 100
 Cube of 11 is: 1331
 Cube of 81 is: 531441
 Square of 36 is : 1296

Q2. Write a java program that displays the number of characters, lines, and words in a text file.

Program:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class FileState{
    public static void main(String args[]){
        String filePath = "ReadFile.txt";
        try{
            FileReader fileReader = new FileReader(filePath);
            BufferedReader bufferedReader = new BufferedReader(fileReader);
            int charCount = 0;
            int lineCount = 0;
            int wordCount = 0;
            String line;
            while((line = bufferedReader.readLine()) != null){
                //characters count
                charCount += line.length();
                //line count
                lineCount++;
                //words count
                String[] words = line.split("\\s+");
                wordCount += words.length;
            }
            System.out.println("Number of characters: " + charCount);
            System.out.println("Number of lines: " + lineCount);
            System.out.println("Number of words: " + wordCount);

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

Output:

Number of characters: 134
 Number of lines: 3
 Number of words: 25

Q3. Write a java program that reads a file and displays the file on the screen with a line number before each line.

Program:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class DisplayLineNumber
{
    public static void main(String args[])
    {
        String filePath = "ReadFile.txt";
        try{
            FileReader fileReader = new FileReader(filePath);
            BufferedReader bufferedReader = new BufferedReader(fileReader);
            String line;
            int lineNumber = 1;
            while((line = bufferedReader.readLine()) != null)
            {
                System.out.println(lineNumber + ": " + line);
                lineNumber++;
            }
            bufferedReader.close();
        } catch(IOException e){
            e.printStackTrace();
        }
    }
}
```

Output:

- 1: This is sumit khowal.
- 2: This is program of reading characters and words and lines from the file.
- 3: and this is assignment of java mca sem-2.

Q4. Write a java program that loads names and phone numbers from a text file where the data is organized as one line per record, and each field in record is separated by a tab (/t). It takes a name or phone number as input and prints the corresponding other value from the hash table. Hint: hash tables.

Program:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Phonebook {
    public static void main(String[] args) {
        String filePath = "phonebook.txt";
        Map<String, String> phonebook = new HashMap<>();

        try {
            FileReader fileReader = new FileReader(filePath);
            BufferedReader bufferedReader = new BufferedReader(fileReader);
            String line;
            while ((line = bufferedReader.readLine()) != null) {
                String[] parts = line.split("\\s+");
                if (parts.length == 2) {
                    String name = parts[0];
                    String phoneNumber = parts[1];
                    phonebook.put(name, phoneNumber);
                }
            }
            bufferedReader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a name or phone number to search: ");
        String input = scanner.nextLine().trim();

        if (phonebook.containsKey(input)) {
            String result = phonebook.get(input);
```

```

        System.out.println("Name: " + input + ", Phone Number: " + result);
    } else if (phonebook.containsValue(input)) {
        for (Map.Entry<String, String> entry : phonebook.entrySet()) {
            if (entry.getValue().equals(input)) {
                String result = entry.getKey();
                System.out.println("Phone Number: " + input + ", Name: " + result);
            }
        }
    } else {
        System.out.println("Entry not found.");
    }
    scanner.close();
}
}

```

Output:

Enter a name or phone number to search: SK
 Name: SK, Phone Number: 555-123-4567

Q5. Write a java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.

Program:

```

package informationdatabase;
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class InformationDatabase
{
    public static void main(String[] args) throws Exception
    {
        int choice = -1;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        DbConnection con = new DbConnection();
        do {
            System.out.println();
            System.out.println("-----");
            System.out.println("1.Insert Data");
            System.out.println("2.Display Data");

```

```

System.out.println("3.Update Data");
System.out.println("4.Delete Data");
System.out.println("5.Exit");
System.out.println("-----");
System.out.println("Enter your choice: ");
choice = Integer.parseInt(br.readLine());
System.out.println();

switch (choice) {
case 1:
    System.out.print("Enter number of input Data: ");
    int n = Integer.parseInt(br.readLine());
    for(int i=0;i<n;i++){
        System.out.print("Enter Roll number: ");
        int rollno = Integer.parseInt(br.readLine());

        System.out.print("Enter Name: ");
        String name =br.readLine();

        System.out.print("Enter Age: ");
        int age = Integer.parseInt(br.readLine());

        System.out.print("Enter Branch: ");
        String branch =br.readLine();

        con.insert(rollno,name,age,branch);
    }
    break;

case 2:
    System.out.println("Roll no.    Name \t\t Age \tBranch ");
    System.out.println("-----");
    con.display();
    break;

case 3:
    System.out.print("Enter Roll number to update: ");
    int rollno = Integer.parseInt(br.readLine());

    System.out.print("Enter new Name: ");
    String name =br.readLine();

    System.out.print("Enter new Age: ");

```

```

        int age = Integer.parseInt(br.readLine());

        System.out.print("Enter new Branch: ");
        String branch = br.readLine();

        con.update(rollno,name,age,branch);
        break;

    case 4:
        System.out.print("Enter Roll number to delete: ");
        rollno = Integer.parseInt(br.readLine());
        con.delete(rollno);
        break;

    case 5:
        con.close();
    }
    } while(choice!=0);
}
}
//creating connection between database(Oracle)
class DbConnection {
    Connection con;
    PreparedStatement ps;
    ResultSet rs;
    DbConnection(){
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","sumit","man
ager");
            System.out.println("Connection Established.");

        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }

    public void insert(int rollno,String name,int age,String branch) throws
SQLException{
        ps =con.prepareStatement("insert into info(rollno,name,age,branch)
values(?,?,?,?)");
        ps.setInt(1, rollno);

```

```

        ps.setString(2, name);
        ps.setInt(3, age);
        ps.setString(4, branch);
        ps.executeUpdate();
        System.out.println("Record Inserted");
    }

    public void display() throws SQLException {
        ps = con.prepareStatement("SELECT * FROM INFO");
        rs = ps.executeQuery();
        while(rs.next()) {
            System.out.printf("%-15s%-20s%-5s%-5s\n", rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4));
        }
    }

    public void update(int rollno, String name, int age, String branch) throws SQLException {
        ps = con.prepareStatement("update info set name=?,age=?,branch=? where rollno=?");

        ps.setInt(4, rollno);
        ps.setString(1, name);
        ps.setInt(2, age);
        ps.setString(3, branch);
        ps.executeUpdate();
        System.out.println("Record Updated.");
    }

    public void delete(int rollno) throws SQLException
    {
        ps = con.prepareStatement("DELETE FROM INFO WHERE ROLLNO = ?");
        ps.setInt(1, rollno);
        ps.executeUpdate();
        System.out.println("Record Deleted.");
    }

    public void close() throws Exception
    {
        con.close();
        System.out.println("Connection closed.");
    }
}

```

Output:

run:

Connection Established.

1.Insert Data
 2.Display Data
 3.Update Data
 4.Delete Data
 5.Exit

Enter your choice: 1

Enter number of input Data: 2

Enter Roll number: 2

Enter Name: Sony

Enter Age: 24

Enter Branch: BCA

Record Inserted

Enter Roll number: 3

Enter Name: Khushboo

Enter Age: 22

Enter Branch: MSC

Record Inserted

1.Insert Data
 2.Display Data
 3.Update Data
 4.Delete Data
 5.Exit

Enter your choice: 2

Roll no.	Name	Age	Branch
----------	------	-----	--------

1	Sumit	23	MCA
---	-------	----	-----

2	Sony	24	BCA
---	------	----	-----

3	Khushboo	22	MSC
---	----------	----	-----

1.Insert Data
 2.Display Data
 3.Update Data
 4.Delete Data
 5.Exit

Enter your choice: 5

Connection closed.