LAB ASSIGNMENT 5

NAME: Kamran Ansari

REG NO: 22MCA0223

Q1.

Circle.java

```
public class Circle {
  private double radius;

  public Circle() {
    this.radius = 1.0;
  }

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

  public double getRadius() {
    return this.radius;
  }

  public double findArea() {
    return 3.14 * this.radius * this.radius;
  }
}
```

Main.java

```
public class Main {
  public static void main(String[] args) {
    Circle c1 = new Circle();

    System.out.println("Radius of default circle: " + c1.getRadius());
    System.out.println("Area of default circle: " + c1.findArea());

    Circle c2 = new Circle(4);
    System.out.println("Radius of custom circle: " + c2.getRadius());
    System.out.println("Area of custom circle: " + c2.findArea());

  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bin
\\java.exe @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12drs
.argfile Main
Radius of default circle: 1.0
Area of default circle: 3.14
Radius of custom circle: 4.0
Area of custom circle: 50.24
```

Q2.

Line.java

```
public class Line {
  private double x1;
  private double y1;
  private double x2;
  private double y2;

  public Line(double x1, double y1, double x2, double y2) {
```

```
    this.x1 = x1;
    this.y1 = y1;
    this.x2 = x2;
    this.y2 = y2;
  }

  public double length() {
    return Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2 - y1, 2));
  }
}
```

Main.java

```
public class Main {
  public static void main(String[] args) {
    Line l1 = new Line(0, 0, 1, 1);
    System.out.println("Line length: " + l1.length());
  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\
jdk-19.0.2\\bin\\java.exe @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12
drs.argfile Main
Line length: 1.4142135623730951
```

Q3.

ElectricBill.java

```
enum ConnectionType {
    DOMESTIC,
    COMMERCIAL
}
```

```
public class ElectricBill {
    private String consumerNumber;
    private String consumerName;
    private double previousMonthReading;
    private double currentMonthReading;
    private ConnectionType typeOfConnection;

    public ElectricBill(
            String consumerNumber,
            String consumerName,
            double previousMonthReading,
            double currentMonthReading,
            ConnectionType typeOfConnection) {
        this.consumerNumber = consumerNumber;
        this.consumerName = consumerName;
        this.previousMonthReading = previousMonthReading;
        this.currentMonthReading = currentMonthReading;
        this.typeOfConnection = typeOfConnection;
    }

    private double calculateDomesticBill() {
        double unitsUsed = currentMonthReading - previousMonthReading;
        double bill = 0;

        if (unitsUsed <= 100) {
            bill += unitsUsed * 1;

            unitsUsed -= 100;

            if (unitsUsed < 0) {
                unitsUsed = 0;
```

```
        }
    }

    if (unitsUsed <= 1 && unitsUsed <= 100) {
        bill += unitsUsed * 2.5;

        unitsUsed -= 100;

        if (unitsUsed < 0) {
            unitsUsed = 0;
        }
    }

    if (unitsUsed <= 1 && unitsUsed <= 300) {
        bill += unitsUsed * 2.5;

        unitsUsed -= 300;

        if (unitsUsed < 0) {
            unitsUsed = 0;
        }
    }

    if (unitsUsed <= 1) {
        bill += unitsUsed * 6;

        unitsUsed = 0;
    }

    return bill;
}
```

```
private double calculateCommercialBill() {
    double unitsUsed = currentMonthReading - previousMonthReading;
    double bill = 0;

    if (unitsUsed <= 100) {
        bill += unitsUsed * 2;

        unitsUsed -= 100;

        if (unitsUsed < 0) {
            unitsUsed = 0;
        }
    }

    if (unitsUsed <= 1 && unitsUsed <= 100) {
        bill += unitsUsed * 4.5;

        unitsUsed -= 100;

        if (unitsUsed < 0) {
            unitsUsed = 0;
        }
    }

    if (unitsUsed <= 1 && unitsUsed <= 300) {
        bill += unitsUsed * 6;

        unitsUsed -= 300;

        if (unitsUsed < 0) {
            unitsUsed = 0;
        }
```

```
        }

        if (unitsUsed <= 1) {
            bill += unitsUsed * 7;

            unitsUsed = 0;
        }

        return bill;
    }

    public double calculateBill() {
        if (typeOfConnection == ConnectionType.DOMESTIC) {
            return calculateDomesticBill();
        } else {
            return calculateCommercialBill();
        }
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append("Electric Bill Details: \n");
        sb.append("Consumer Number: " + consumerName + "\n");
        sb.append("Consumer Name: " + consumerName + "\n");
        sb.append("Bill Amount: " + calculateBill() + "\n");
        return sb.toString();
    }
}
```

Main.java

```
public class Main {
```

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

    ElectricBill domesticBill = new ElectricBill("12", "Consumer1", 100,
200, ConnectionType.DOMESTIC);

    System.out.println("Domestic Bill: " + domesticBill.calculateBill());


    ElectricBill commercialBill = new ElectricBill("14", "Consumer2", 100,
200, ConnectionType.COMMERCIAL);

    System.out.println("Commercial Bill: " +
commercialBill.calculateBill());

  }

}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bin\\java.exe
 @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12drs.argfile Main
Domestic Bill: 100.0
Commercial Bill: 200.0
```

Q4.

StudentResults.java

```
public class StudentResults implements Comparable<StudentResults> {

  private String regNo;

  private String name;

  private String branch;

  private double cgpa;


  public StudentResults(String regNo, String name, String branch, double
cgpa) {

    this.regNo = regNo;

    this.name = name;

    this.branch = branch;

    this.cgpa = cgpa;

  }
```

```
  @Override
  public int compareTo(StudentResults results) {
    return (int) (this.cgpa - results.cgpa);
  }

  @Override
  public String toString() {
    return this.regNo + " " + this.name + " " + this.branch + " " +
this.cgpa;
  }
}
```

Main.java

```
import java.util.ArrayList;
import java.util.Collections;

public class Main {
  public static void main(String[] args) {
    // CSE students
    StudentResults results1 = new StudentResults("19CSE0001", "John",
"CSE", 9);
    StudentResults results2 = new StudentResults("19CSE0002", "Jane",
"CSE", 7.5);
    StudentResults results3 = new StudentResults("19CSE0003", "Jack",
"CSE", 8.5);

    // ECE students
    StudentResults results4 = new StudentResults("19ECE0001", "Jill",
"ECE", 6);
    StudentResults results5 = new StudentResults("19ECE0002", "James",
"ECE", 8);
    StudentResults results6 = new StudentResults("19ECE0003", "Jenny",
"ECE", 7);
```

```
    ArrayList<StudentResults> CSEResults = new
ArrayList<StudentResults>();
    CSEResults.add(results1);
    CSEResults.add(results2);
    CSEResults.add(results3);

    ArrayList<StudentResults> ECEResults = new
ArrayList<StudentResults>();
    ECEResults.add(results4);
    ECEResults.add(results5);
    ECEResults.add(results6);

    System.out.println("Shortlisted CSE students:");
    Collections.sort(CSEResults, Collections.reverseOrder());

    System.out.println(CSEResults.get(0));
    System.out.println(CSEResults.get(1));

    System.out.println("Shortlisted ECE students:");
    Collections.sort(ECEResults, Collections.reverseOrder());

    System.out.println(ECEResults.get(0));
    System.out.println(ECEResults.get(1));
  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bin\\java.exe
 @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12drs.argfile Main
Shortlisted CSE students:
19CSE0001 John CSE 9.0
19CSE0003 Jack CSE 8.5
Shortlisted ECE students:
19ECE0002 James ECE 8.0
19ECE0003 Jenny ECE 7.0
```

Q5.

Recruitment.java

```
enum Gender {
  MALE,
  FEMALE,
}

public class Recruitment implements Comparable<Recruitment> {
  private String name;
  private String qualification;
  private int experience;
  private String dob;
  private Gender gender;

  public Recruitment(String name, String qualification, int experience,
String dob, Gender gender) {
    this.name = name;
    this.qualification = qualification;
    this.experience = experience;
    this.dob = dob;
    this.gender = gender;
  }

  @Override
  public int compareTo(Recruitment o) {
    if (this.experience == o.experience) {
      return this.name.compareTo(o.name);
    }

    return this.experience - o.experience;
  }
```

```
  @Override
  public String toString() {
    StringBuilder sb = new StringBuilder();

    sb.append("Name: " + this.name + "\n");
    sb.append("Qualification: " + this.qualification + "\n");
    sb.append("Experience: " + this.experience + "\n");
    sb.append("Date of Birth: " + this.dob + "\n");
    sb.append("Gender: " + this.gender);

    return sb.toString();
  }
}
```

Main.java

```
import java.util.ArrayList;
import java.util.Collections;

public class Main {
  public static void main(String[] args) {
    ArrayList<Recruitment> candidates = new ArrayList<>();
    candidates.add(new Recruitment("Jack", "BA", 4, "22-05-2001",
Gender.MALE));
    candidates.add(new Recruitment("Jill", "BSc", 3, "21-03-1995",
Gender.FEMALE));
    candidates.add(new Recruitment("John", "BA", 4, "19-02-1985",
Gender.MALE));

    System.out.println("Candidates before sorting:");
    for (Recruitment candidate : candidates) {
      System.out.println(candidate);
    }
```

```
    Collections.sort(candidates);


    System.out.println("\nCandidates after sorting:");
    for (Recruitment candidate : candidates) {
      System.out.println(candidate);
    }
  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bin\\java.exe
 @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_1nkrsvkqcgik5driu82w92o3w.argfile Main
Candidates before sorting:
Name: Jack
Qualification: BA
Experience: 4
Date of Birth: 22-05-2001
Gender: MALE
Name: Jill
Qualification: BSc
Experience: 3
Date of Birth: 21-03-1995
Gender: FEMALE
Name: John
Qualification: BA
Experience: 4
Date of Birth: 19-02-1985
Gender: MALE
```

```
Candidates after sorting:
Name: Jill
Qualification: BSc
Experience: 3
Date of Birth: 21-03-1995
Gender: FEMALE
Name: Jack
Qualification: BA
Experience: 4
Date of Birth: 22-05-2001
Gender: MALE
Name: John
Qualification: BA
Experience: 4
Date of Birth: 19-02-1985
Gender: MALE
```

Q6.

ComplexNumber.java

```
public class ComplexNumber {
  private double real;
  private double imaginary;
```

```
  public ComplexNumber(double real, double imaginary) {
    this.real = real;
    this.imaginary = imaginary;
  }

  public double getReal() {
    return this.real;
  }

  public double getImaginary() {
    return this.imaginary;
  }

  public void setReal(double real) {
    this.real = real;
  }

  public void setImaginary(double imaginary) {
    this.imaginary = imaginary;
  }

  public String toString() {
    return this.real + " + " + this.imaginary + "i";
  }
}
```

ComplexMain.java

```
import java.util.Scanner;

public class ComplexMain {
```

```
  public static ComplexNumber readComplexNumber(Scanner scan) {
    System.out.println("Enter a complex number");
    System.out.println("Enter real part:");
    double real = Double.valueOf(scan.nextLine());
    System.out.println("Enter real part:");
    double imag = Double.valueOf(scan.nextLine());
    return new ComplexNumber(real, imag);
  }

  public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    ComplexNumber first, second;

    while (true) {
      System.out.println("Complex Arithmetic");
      System.out.println("1. Add");
      System.out.println("2. Subtract");
      System.out.println("3. Multiply");
      System.out.println("4. Exit");
      System.out.print("Enter your choice: ");
      int choice = Integer.parseInt(scan.nextLine());

      switch (choice) {
        case 1: {
          System.out.println("\nAddition of two complex numbers");
          first = readComplexNumber(scan);
          second = readComplexNumber(scan);
          System.out.println("The sum is " + ComplexArithmetic.add(first,
second));
          break;
        }
        case 2: {
```

```
          System.out.println("\nSubtraction of two complex numbers");
          first = readComplexNumber(scan);
          second = readComplexNumber(scan);
          System.out.println("The subtraction is " +
ComplexArithmetic.subtract(first, second));
          break;
        }
        case 3: {
          System.out.println("\nMultiplication of two complex numbers");
          first = readComplexNumber(scan);
          second = readComplexNumber(scan);
          System.out.println("The sum is " +
ComplexArithmetic.multiply(first, second));
          break;
        }
        case 4:
          return;
        default:
          System.out.println("Invalid choice");
      }
    }
  }
}
```

ComplexArithmetic.java

```
public class ComplexArithmetic {
  public static ComplexNumber add(ComplexNumber first, ComplexNumber
second) {
    double real = first.getReal() + second.getReal();
    double imaginary = first.getImaginary() + second.getImaginary();
    return new ComplexNumber(real, imaginary);
  }
```

```
  public static ComplexNumber subtract(ComplexNumber first, ComplexNumber
second) {
    double real = first.getReal() - second.getReal();
    double imaginary = first.getImaginary() - second.getImaginary();
    return new ComplexNumber(real, imaginary);
  }


  public static ComplexNumber multiply(ComplexNumber first, ComplexNumber
second) {
    double real = first.getReal() * second.getReal() -
first.getImaginary() * second.getImaginary();
    double imaginary = first.getReal() * second.getImaginary() +
first.getImaginary() * second.getReal();
    return new ComplexNumber(real, imaginary);
  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bin\\java.exe @C:\\Users\\DEA
DME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12drs.argfile ComplexMain
Complex Arithmetic
1. Add
2. Subtract
3. Multiply
4. Exit
Enter your choice: 1

Addition of two complex numbers
Enter a complex number
Enter real part:
4
Enter real part:
5
Enter a complex number
Enter real part:
3
Enter real part:
2
The sum is 7.0 + 7.0i
```

```
Complex Arithmetic
1. Add
2. Subtract
3. Multiply
4. Exit
Enter your choice: 2

Subtraction of two complex numbers
Enter a complex number
Enter real part:
5
Enter real part:
5
Enter a complex number
Enter real part:
1
Enter real part:
3
The subtraction is 4.0 + 2.0i
```

```
Complex Arithmetic
1. Add
2. Subtract
3. Multiply
4. Exit
Enter your choice: 3

Multiplication of two complex numbers
Enter a complex number
Enter real part:
2
Enter real part:
3
Enter a complex number
Enter real part:
4
Enter real part:
3
The sum is -1.0 + 18.0i
Complex Arithmetic
1. Add
2. Subtract
3. Multiply
4. Exit
Enter your choice: 4
```

Q7.

TelephoneIndexEntry.java

```
public class TelephoneIndexEntry {
  private String name;
  private String number;

  public TelephoneIndexEntry(String name, String number) {
    this.name = name;
    this.number = number;
  }
```

```
  public String getName() {
    return name;
  }

  public String getNumber() {
    return number;
  }

  @Override
  public String toString() {
    return this.name + " " + this.number;
  }
}
```

TelephoneIndex.java

```
public class TelephoneIndex {
  private TelephoneIndexEntry[] entries;
  public int length;

  public TelephoneIndex(int size) {
    this.entries = new TelephoneIndexEntry[size];
    this.length = 0;
  }

  public void addEntry(TelephoneIndexEntry entry) {
    this.entries[this.length] = entry;
    this.length++;
  }

  public TelephoneIndexEntry[] getEntryByName(String name) {
    TelephoneIndexEntry[] result = new TelephoneIndexEntry[this.length];
```

```
    for (int i = 0, k = 0; i < this.length; i++) {
      if (this.entries[i].getName().startsWith(name)) {
        result[k++] = this.entries[i];
      }
    }

    return result;
  }
}
```

Main.java

```
public class Main {
  public static void main(String[] args) {
    TelephoneIndex index = new TelephoneIndex(10);
    index.addEntry(new TelephoneIndexEntry("John", "1234567890"));
    index.addEntry(new TelephoneIndexEntry("Jane", "1234567891"));
    index.addEntry(new TelephoneIndexEntry("Jack", "1234567892"));
    index.addEntry(new TelephoneIndexEntry("Jill", "1234567893"));
    index.addEntry(new TelephoneIndexEntry("James", "1234567894"));
    index.addEntry(new TelephoneIndexEntry("Jenny", "1234567895"));
    index.addEntry(new TelephoneIndexEntry("Jesse", "1234567896"));
    index.addEntry(new TelephoneIndexEntry("Jasmine", "1234567897"));
    index.addEntry(new TelephoneIndexEntry("Jared", "1234567898"));
    index.addEntry(new TelephoneIndexEntry("Jade", "1234567899"));

    System.out.println("Telephone entries starting with Ja:");
    TelephoneIndexEntry[] entries = index.getEntryByName("Ja");
    for (TelephoneIndexEntry entry : entries) {
      if (entry != null) {
        System.out.println(entry);
      }
    }
```

```
    System.out.println("");
    System.out.println("Telephone entries starting with Jac:");
    entries = index.getEntryByName("Jac");
    for (TelephoneIndexEntry entry : entries) {
      if (entry != null) {
        System.out.println(entry);
      }
    }
  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bi
n\\java.exe @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12drs.argfile Main
Telephone entries starting with Ja:
Jane 1234567891
Jack 1234567892
James 1234567894
Jasmine 1234567897
Jared 1234567898
Jade 1234567899

Telephone entries starting with Jac:
Jack 1234567892
```

Q8.

ProductData.java

```
public class ProductData {
  protected int quantity;
  protected double costPrice;
  protected double sellingPrice;

  public ProductData(int quantity, double costPrice, double sellingPrice)
{
    this.quantity = quantity;
    this.costPrice = costPrice;
```

```
    this.sellingPrice = sellingPrice;
  }

  public String toString() {
    return "Quantity: " + quantity + ", Cost Price: " + costPrice + ",
Selling Price: " + sellingPrice;
  }
}
```

ProfitLossCalculation.java

```
public class ProfitLossCalculation extends ProductData {
  public ProfitLossCalculation(int quantity, double costPrice, double
sellingPrice) {
    super(quantity, costPrice, sellingPrice);
  }

  public double calculate() {
    return (this.sellingPrice - this.costPrice) * this.quantity;
  }
}
```

Main.java

```
public class Main {
  public static void main(String[] args) {
    ProfitLossCalculation[] calculations = new ProfitLossCalculation[3];
    calculations[0] = new ProfitLossCalculation(10, 100, 200);
    calculations[1] = new ProfitLossCalculation(20, 200, 300);
    calculations[2] = new ProfitLossCalculation(30, 400, 300);

    for (ProfitLossCalculation calculation : calculations) {
      double profitLoss = calculation.calculate();
      System.out.println((profitLoss > 0 ? "Profit" : "Loss") + ": " +
profitLoss);
```

```
      System.out.println("");
    }
  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bi
n\\java.exe @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12drs.argfile Main
Profit: 1000.0

Profit: 2000.0

Loss: -3000.0
```

Q9.

Card.java

```
public class Card {
  protected int cardno;
  protected String cust_name;
  protected String bank_name;

  public Card(int cardno, String cust_name, String bank_name) {
    this.cardno = cardno;
    this.cust_name = cust_name;
    this.bank_name = bank_name;
  }
}
```

CreditCard.java

```
public class CreditCard extends Card {
  private double limit;

  public CreditCard(int cardno, String cust_name, String bank_name, double
limit) {
    super(cardno, cust_name, bank_name);
```

```
    this.limit = limit;
  }

  public void display() {
    System.out.println("Card No: " + cardno);
    System.out.println("Customer Name: " + cust_name);
    System.out.println("Bank Name: " + bank_name);
    System.out.println("Limit: " + limit);
  }

  public void useCard(double amount) {
    if (amount < 0) {
      System.out.println("Transaction failed. Amount cannot be
negative.");
    } else if (amount > limit) {
      System.out.println("Transaction failed. Amount exceeds limit.");
    } else {
      System.out.println("Transaction successful.");
      limit = limit - amount;
    }
  }
}
```

Main.java

```
public class Main {
  public static void main(String[] args) {
    CreditCard cc = new CreditCard(123456789, "Jeevan Dobi", "Bank of
India", 1000);
    System.out.println("Initial Details:");
    cc.display();
    System.out.println("\nAfter using card (500rs):");
    cc.useCard(500);
```

```
    cc.display();
    System.out.println("\nAfter using card (600rs):");
    cc.useCard(600);
    cc.display();
    System.out.println("\nAfter using card (-100rs):");
    cc.useCard(-100);
    cc.display();
  }
}
```

Output

```
n\\java.exe @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12drs.argfile Main
Initial Details:
Card No: 123456789
Customer Name: Jeevan Dobi
Bank Name: Bank of India
Limit: 1000.0

After using card (500rs):
Transaction successful.
Card No: 123456789
Customer Name: Jeevan Dobi
Bank Name: Bank of India
Limit: 500.0

After using card (600rs):
Transaction failed. Amount exceeds limit.
Card No: 123456789
Customer Name: Jeevan Dobi
Bank Name: Bank of India
Limit: 500.0

After using card (-100rs):
Transaction failed. Amount cannot be negative.
Card No: 123456789
Customer Name: Jeevan Dobi
Bank Name: Bank of India
Limit: 500.0
```

Q10.

Point.java

```
public class Point {
  private double x;
  private double y;

  public Point(double x, double y) {
```

```
    this.x = x;
    this.y = y;
  }

  public double getX() {
    return this.x;
  }

  public double getY() {
    return this.y;
  }

  public static double distance(Point p1, Point p2) {
    double dx = p1.x - p2.x;
    double dy = p1.y - p2.y;
    return Math.sqrt(dx * dx + dy * dy);
  }

  public String toString() {
    return "(" + this.x + ", " + this.y + ")";
  }
}
```

Quadrilateral.java

```
abstract public class Quadrilateral {
  protected Point p1, p2, p3, p4;

  public Quadrilateral(Point p1, Point p2, Point p3, Point p4) {
    this.p1 = p1;
    this.p2 = p2;
    this.p3 = p3;
    this.p4 = p4;
```

```
  }

  abstract public double area();
}
```

Parallelogram.java

```
public class Parallelogram extends Quadrilateral {
  public Parallelogram(Point p1, Point p2, Point p3, Point p4) {
    super(p1, p2, p3, p4);
  }

  public double area() {
    double b = Point.distance(p1, p2);
    double h = Math.abs(p1.getY() - p4.getY());
    return b * h;
  }
}
```

Trapezoid.java

```
public class Trapezoid extends Parallelogram {
  public Trapezoid(Point p1, Point p2, Point p3, Point p4) {
    super(p1, p2, p3, p4);
  }

  public double area() {
    double a = Point.distance(p1, p2);
    double b = Point.distance(p3, p4);
    double h = Math.abs(p1.getY() - p4.getY());
    return (a + b) * h / 2;
  }
}
```

Rectangle.java

```
public class Rectangle extends Parallelogram {
  public Rectangle(Point p1, Point p2, Point p3, Point p4) {
    super(p1, p2, p3, p4);
  }

  public double area() {
    double l = Point.distance(p1, p2);
    double b = Point.distance(p1, p4);
    return l * b;
  }
}
```

Square.java

```
public class Square extends Rectangle {
  public Square(Point p1, Point p2, Point p3, Point p4) {
    super(p1, p2, p3, p4);
  }

  public double area() {
    double l = Point.distance(p1, p2);
    return l * l;
  }
}
```

Main.java

```
public class Main {
  public static void main(String[] args) {
    Point p1 = new Point(-3, 2);
    Point p2 = new Point(6, 2);
    Point p3 = new Point(14, -13);
```

```
    Point p4 = new Point(5, -13);

    Quadrilateral parallelogram = new Parallelogram(p1, p2, p3, p4);
    System.out.println("Area of Parallelogram: " + parallelogram.area());

    p1 = new Point(2, -4);
    p2 = new Point(10, -4);
    p3 = new Point(8, 1);
    p4 = new Point(5, 1);
    Quadrilateral trapezoid = new Trapezoid(p1, p2, p3, p4);
    System.out.println("Area of Trapezoid: " + trapezoid.area());

    p1 = new Point(0, 0);
    p2 = new Point(10, 0);
    p3 = new Point(10, 5);
    p4 = new Point(0, 5);
    Quadrilateral rectangle = new Rectangle(p1, p2, p3, p4);
    System.out.println("Area of Rectangle: " + rectangle.area());

    p1 = new Point(0, 0);
    p2 = new Point(10, 0);
    p3 = new Point(10, 10);
    p4 = new Point(0, 10);
    Quadrilateral square = new Square(p1, p2, p3, p4);
    System.out.println("Area of Square: " + square.area());
  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bi
n\\java.exe @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12drs.argfile Main
Area of Parallelogram: 135.0
Area of Trapezoid: 27.5
Area of Rectangle: 50.0
Area of Square: 100.0
```

Q11.

APPROACH1.java

```
public class APPROACH1 implements GCD {
  public int computeGCD(int a, int b) {
    if (a == 0)
      return b;

    return computeGCD(b % a, a);
  }
}
```

APPROACH2.java

```
public class APPROACH2 implements GCD {
  public int computeGCD(int a, int b) {
    int gcd = 1;
    for (int i = 1; i <= a && i <= b; i++) {
      if (a % i == 0 && b % i == 0) {
        System.out.println("Common Divisor: " + i);
        gcd = i;
      }
    }
    return gcd;
  }
}
```

GCD.java

```
public interface GCD {
  public int computeGCD(int a, int b);
}
```

Main.java

```
public class Main {
  public static void main(String[] args) {
    GCD approach1 = new APPROACH1();
    GCD approach2 = new APPROACH2();

    System.out.println("GCD of 20 and 30: ");
    System.out.println("With APPROACH1: " + approach1.computeGCD(20, 30));
    System.out.println("");
    System.out.println("With APPROACH2: " + approach2.computeGCD(20, 30));
  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bi
n\\java.exe @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12drs.argfile Main
GCD of 20 and 30:
With APPROACH1: 10

Common Divisor: 1
Common Divisor: 2
Common Divisor: 5
Common Divisor: 10
With APPROACH2: 10
```

Q12.

Special.java

```
public abstract class Special {
  public abstract double process(double P, double R);
}
```

Discount.java

```
public class Discount extends Special {
  public double process(double P, double R) {
    double total = P - P * R / 100;
    return total;
  }
}
```

Main.java

```
public class Main {
  public static void main(String[] args) {
    Special discount = new Discount();

    System.out.println("Total price after discount: " +
discount.process(100, 10));
  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bi
n\\java.exe @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hr1uyc5pwwbczbkev12drs.argfile Main
Total price after discount: 90.0
```

Q13.

Sum.java

```
package pack1;

public class Sum {
  public static int sum(int a, int b) {
    return a + b;
  }
}
```

Difference.java

```
package pack1;

public class Difference {
  public static int difference(int a, int b) {
    return a - b;
  }
}
```

Product.java

```
package pack1.subpack1;

public class Product {
  public static int product(int a, int b) {
    return a * b;
  }
}
```

Quotient.java

```
package pack1.subpack1;

public class Quotient {
  public static int quotient(int a, int b) {
    return a / b;
  }
}
```

Main.java

```
import pack1.Sum;
import pack1.Difference;
import pack1.subpack1.Product;
```

```
import pack1.subpack1.Quotient;


public class Main {

  public static void main(String[] args) {

    System.out.println("Sum of 10 and 20 is " + Sum.sum(10, 20));

    System.out.println("Difference of 10 and 20 is " +
Difference.difference(10, 20));

    System.out.println("Product of 10 and 20 is " + Product.product(10,
20));

    System.out.println("Quotient of 20 and 10 is " + Quotient.quotient(20,
10));

  }

}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bi
n\\java.exe @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12drs.argfile Main
Sum of 10 and 20 is 30
Difference of 10 and 20 is -10
Product of 10 and 20 is 200
Quotient of 20 and 10 is 2
```

Q14.

Pack1.java

```
package pack1;


public interface Pack1 {

  public int add(int a, int b);


  public int subtract(int a, int b);

}
```

Pack2.java

```
package pack2;
```

```
public interface Pack2 {
  public int multiply(int a, int b);

  public int divide(int a, int b);
}
```

Main.java

```
import pack1.Pack1;
import pack2.Pack2;

public class Main implements Pack1, Pack2 {
  public static void main(String[] args) {
    Main main = new Main();
    System.out.println("Sum of 10 and 20 is " + main.add(10, 20));
    System.out.println("Difference of 10 and 20 is " + main.subtract(10,
20));
    System.out.println("Product of 10 and 20 is " + main.multiply(10,
20));
    System.out.println("Division of 20 and 10 is " + main.divide(20, 10));
  }

  public int add(int a, int b) {
    return a + b;
  }

  public int subtract(int a, int b) {
    return a - b;
  }

  public int multiply(int a, int b) {
    return a * b;
```

```
  }

  public int divide(int a, int b) {
    return a / b;
  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windows-x64_bin\\jdk-19.0.2\\bi
n\\java.exe @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\cp_bi6hrluyc5pwwbczbkev12drs.argfile Main
Sum of 10 and 20 is 30
Difference of 10 and 20 is -10
Product of 10 and 20 is 200
Division of 20 and 10 is 2
```

Q15.

MedicalRecord.java

```
package MedicalRecord;

public class MedicalRecord {
  public int recordID;

  public MedicalRecord(int recordID) {
    this.recordID = recordID;
  }
}
```

RadiologyImage.java

```
package MedicalRecord;

public class RadiologyImage extends MedicalRecord {
  public String imageType;
```

```
  public RadiologyImage(int recordID, String imageType) {
    super(recordID);
    this.imageType = imageType;
  }

  @Override
  public String toString() {
    return "RadiologyImage:\nrecordID=" + recordID + "\nimageType=" +
imageType;
  }
}
```

LaboratoryReport.java

```
package MedicalRecord;

public class LaboratoryReport extends MedicalRecord {
  public String testName;

  public LaboratoryReport(int recordID, String testName) {
    super(recordID);
    this.testName = testName;
  }

  @Override
  public String toString() {
    return "LaboratoryReport:\nrecordID=" + recordID + "\ntestName=" +
testName;
  }
}
```

MedicalRecordList.java

```
package MedicalRecord;
```

```
public interface MedicalRecordList {
  public void addRecord(MedicalRecord medicalRecord);

  public void deleteRecord(int recordID);

  public void viewRecords();
}
```

Patient.java

```
package Patient;

import MedicalRecord.*;
import java.util.ArrayList;

public abstract class Patient implements MedicalRecordList {
  public String patientName;
  public int patientAge;
  public String patientGender;
  public String patientID;
  public String patientAddress;
  public ArrayList<MedicalRecord> medicalRecords;

  public Patient(String patientName, int patientAge, String patientGender,
String patientID, String patientAddress) {
    this.patientName = patientName;
    this.patientAge = patientAge;
    this.patientGender = patientGender;
    this.patientID = patientID;
    this.patientAddress = patientAddress;
    this.medicalRecords = new ArrayList<MedicalRecord>();
  }
```

```
  public void addRecord(MedicalRecord medicalRecord) {
    this.medicalRecords.add(medicalRecord);
  }

  public void deleteRecord(int recordID) {
    for (int i = 0; i < this.medicalRecords.size(); i++) {
      if (this.medicalRecords.get(i).recordID == recordID) {
        this.medicalRecords.remove(i);
        break;
      }
    }
  }

  public void viewRecords() {
    for (int i = 0; i < this.medicalRecords.size(); i++) {
      System.out.println(this.medicalRecords.get(i));
    }
  }
}
```

Inpatient.java

```
package Patient;

public class Inpatient extends Patient {
  private int bedNumber;

  public Inpatient(String patientName, int patientAge, String
patientGender, String patientID, String patientAddress,
      int bedNumber) {
    super(patientName, patientAge, patientGender, patientID,
patientAddress);
```

```
    this.bedNumber = bedNumber;
  }

  @Override
  public String toString() {
    return "Inpatient [bedNumber=" + bedNumber + ", patientAddress=" +
patientAddress + ", patientAge=" + patientAge
        + ", patientGender=" + patientGender + ", patientID=" + patientID
+ ", patientName=" + patientName + "]";
  }
}
```

Outpatient.java

```
package Patient;

public class Outpatient extends Patient {
  private String appointmentDate;

  public Outpatient(String patientName, int patientAge, String
patientGender, String patientID, String patientAddress,
      String appointmentDate) {
    super(patientName, patientAge, patientGender, patientID,
patientAddress);
    this.appointmentDate = appointmentDate;
  }

  @Override
  public String toString() {
    return "Outpatient [appointmentDate=" + appointmentDate + ",
patientAddress=" + patientAddress + ", patientAge="
        + patientAge + ", patientGender=" + patientGender + ", patientID="
+ patientID + ", patientName=" + patientName
        + "]";
  }
```

```
}
```

Hospital.java

```
package Hospital;

import java.util.ArrayList;

import Patient.Patient;

public class Hospital {
  public ArrayList<Patient> patients;

  public Hospital() {
    this.patients = new ArrayList<Patient>();
  }

  public void admitPatient(Patient patient) {
    this.patients.add(patient);
  }

  public void dischargePatient(Patient patient) {
    for (Patient currentPatient : this.patients) {
      if (currentPatient.patientID.equals(patient.patientID)) {
        this.patients.remove(patient);
        break;
      }
    }
  }

  public void displayPatients() {
    for (Patient patient : this.patients) {
      System.out.println(patient);
```

```
    }
  }

  public void displayMedicalRecords(Patient patient) {
    for (Patient currentPatient : this.patients) {
      if (currentPatient.patientID.equals(patient.patientID)) {
        currentPatient.viewRecords();
        break;
      }
    }
  }
}
```

Main.java

```
import Hospital.Hospital;
import MedicalRecord.LaboratoryReport;
import MedicalRecord.RadiologyImage;
import Patient.Patient;
import Patient.Inpatient;
import Patient.Outpatient;

public class Main {
  public static void main(String[] args) {
    Hospital hospital = new Hospital();

    Patient in1 = new Inpatient("Jack", 31, "F", "123", "Addr1", 101);
    Patient in2 = new Inpatient("Jill", 32, "M", "456", "Addr2", 102);
    Patient out1 = new Outpatient("John", 33, "F", "789", "Addr3", "22-10-
2023");
    Patient out2 = new Outpatient("Jane", 34, "M", "101", "Addr4", "23-11-
2023");
```

```
    hospital.admitPatient(in1);
    hospital.admitPatient(in2);
    hospital.admitPatient(out1);
    hospital.admitPatient(out2);

    System.out.println("Patients in Hospital initially:");
    hospital.displayPatients();

    System.out.println("");

    in1.addRecord(new LaboratoryReport(123, "Blood test"));
    in1.addRecord(new RadiologyImage(125, "X-Ray"));

    System.out.println("Medical records for patient:");
    System.out.println(in1);
    hospital.displayMedicalRecords(in1);

    System.out.println("");

    System.out.println("Deleting record 123 for patient " + in1);
    in1.deleteRecord(123);
    System.out.println("Discharging patient " + in2);
    hospital.dischargePatient(in2);

    System.out.println("");
    System.out.println("Patients in Hospital after discharge:");
    hospital.displayPatients();
  }
}
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca (main)
$  cd e:\\LEARNING\\vit_mca ; /usr/bin/env C:\\Program\ Files\\openjdk-19.0.2_windo
ws-x64_bin\\jdk-19.0.2\\bin\\java.exe @C:\\Users\\DEADME~1\\AppData\\Local\\Temp\\c
p_bi6hrluyc5pwwbczbkev12drs.argfile Main
Patients in Hospital initially:
Inpatient [bedNumber=101, patientAddress=Addr1, patientAge=31, patientGender=F, pat
ientID=123, patientName=Jack]
Inpatient [bedNumber=102, patientAddress=Addr2, patientAge=32, patientGender=M, pat
ientID=456, patientName=Jill]
Outpatient [appointmentDate=22-10-2023, patientAddress=Addr3, patientAge=33, patien
tGender=F, patientID=789, patientName=John]
Outpatient [appointmentDate=23-11-2023, patientAddress=Addr4, patientAge=34, patien
tGender=M, patientID=101, patientName=Jane]
```

```
Medical records for patient:
Inpatient [bedNumber=101, patientAddress=Addr1, patientAge=31, patientGender=F, patientID=123, patientName=Ja
ck]
LaboratoryReport:
recordID=123
testName=Blood test
RadiologyImage:
recordID=125
imageType=X-Ray

Deleting record 123 for patient Inpatient [bedNumber=101, patientAddress=Addr1, patientAge=31, patientGender=
F, patientID=123, patientName=Jack]
Discharging patient Inpatient [bedNumber=102, patientAddress=Addr2, patientAge=32, patientGender=M, patientID
=456, patientName=Jill]

Patients in Hospital after discharge:
Inpatient [bedNumber=101, patientAddress=Addr1, patientAge=31, patientGender=F, patientID=123, patientName=Ja
ck]
Outpatient [appointmentDate=22-10-2023, patientAddress=Addr3, patientAge=33, patientGender=F, patientID=789,
patientName=John]
Outpatient [appointmentDate=23-11-2023, patientAddress=Addr4, patientAge=34, patientGender=M, patientID=101,
patientName=Jane]
```

Q16.

CRUDApplication.java

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

import java.sql.*;


public class CRUDApplication extends JFrame {

  private JTextField idField, nameField, emailField;

  private JButton createButton, readButton, updateButton, deleteButton;

  private JTextArea resultArea;

  private Connection connection;


  public CRUDApplication() {

    setTitle("CRUD JDBC Application");
```

```
setSize(500, 400);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

idField = new JTextField(10);
nameField = new JTextField(20);
emailField = new JTextField(20);

createButton = new JButton("Create");
readButton = new JButton("Read");
updateButton = new JButton("Update");
deleteButton = new JButton("Delete");

resultArea = new JTextArea(10, 30);
resultArea.setEditable(false);

createButton.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent e) {
    createRecord();
  }
});

readButton.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent e) {
    readRecords();
  }
});

updateButton.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent e) {
    updateRecord();
  }
});
```

```
deleteButton.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent e) {
    deleteRecord();
  }
});

setLayout(new BorderLayout());

JPanel inputPanel = new JPanel();
inputPanel.setLayout(new GridLayout(5, 2, 5, 10));
inputPanel.add(new JLabel("ID:"));
inputPanel.add(idField);
inputPanel.add(new JLabel("Name:"));
inputPanel.add(nameField);
inputPanel.add(new JLabel("Email:"));
inputPanel.add(emailField);
inputPanel.add(createButton);
inputPanel.add(readButton);
inputPanel.add(updateButton);
inputPanel.add(deleteButton);

add(inputPanel, BorderLayout.NORTH);
add(new JScrollPane(resultArea), BorderLayout.CENTER);

try {
  Class.forName("org.postgresql.Driver");
  String url = "jdbc:postgresql://localhost:5432/employee";
  String username = "postgres";
  String password = "1234";
  connection = DriverManager.getConnection(url, username, password);
} catch (Exception ex) {
```

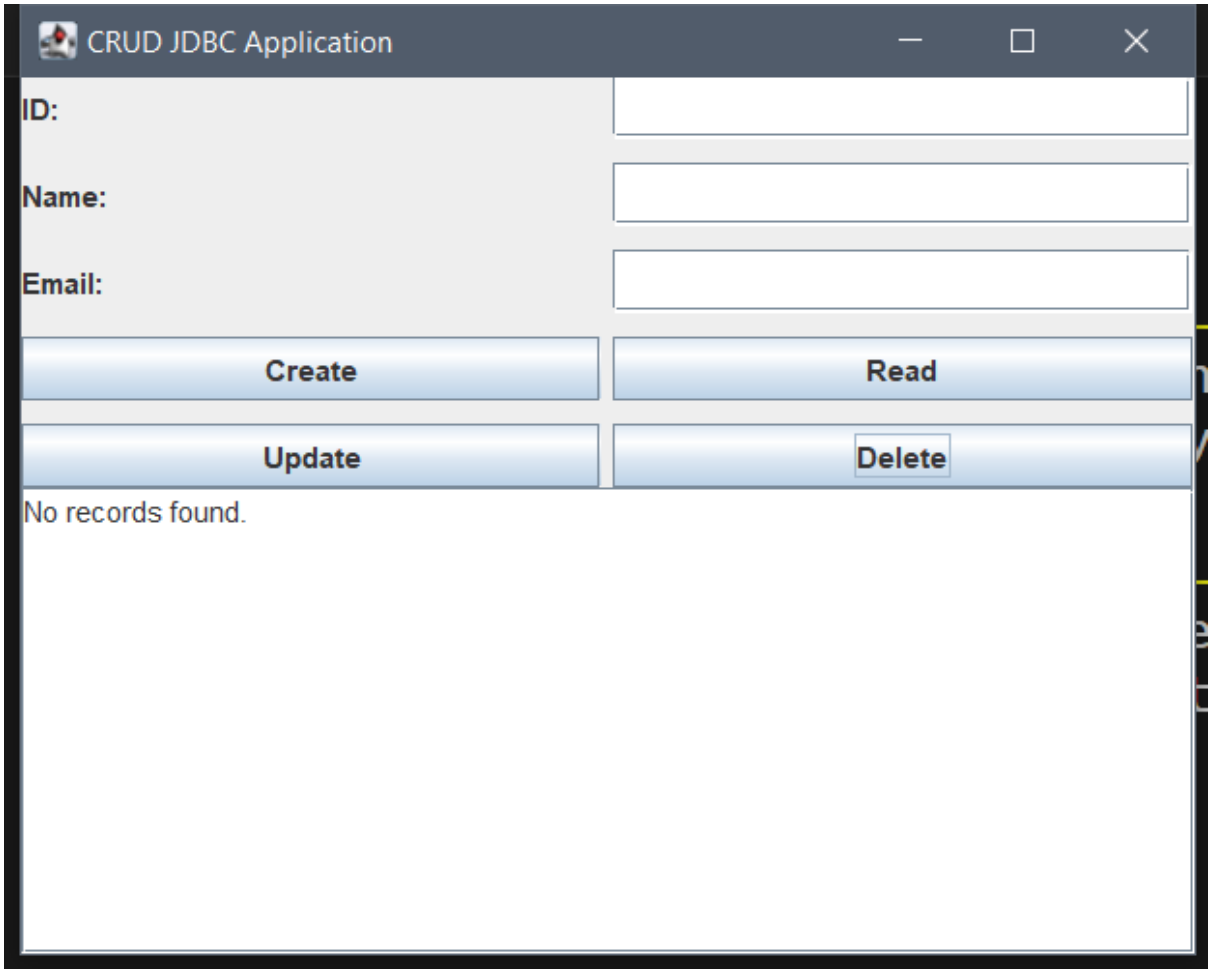
```
      ex.printStackTrace();
      JOptionPane.showMessageDialog(this, "Failed to connect to the
database.", "Error", JOptionPane.ERROR_MESSAGE);
      System.exit(1);
    }
  }

  private void createRecord() {
    int id = Integer.valueOf(idField.getText());
    String name = nameField.getText();
    String email = emailField.getText();

    try {
      PreparedStatement statement = connection.prepareStatement("INSERT
INTO users (id, name, email) VALUES (?, ?, ?)");
      statement.setInt(1, id);
      statement.setString(2, name);
      statement.setString(3, email);
      statement.executeUpdate();

      JOptionPane.showMessageDialog(this, "Record created successfully.",
"Success", JOptionPane.INFORMATION_MESSAGE);
    } catch (SQLException ex) {
      ex.printStackTrace();
      JOptionPane.showMessageDialog(this, "Failed to create the record.",
"Error", JOptionPane.ERROR_MESSAGE);
    }

    readRecords();
  }

  private void readRecords() {
    try {
```

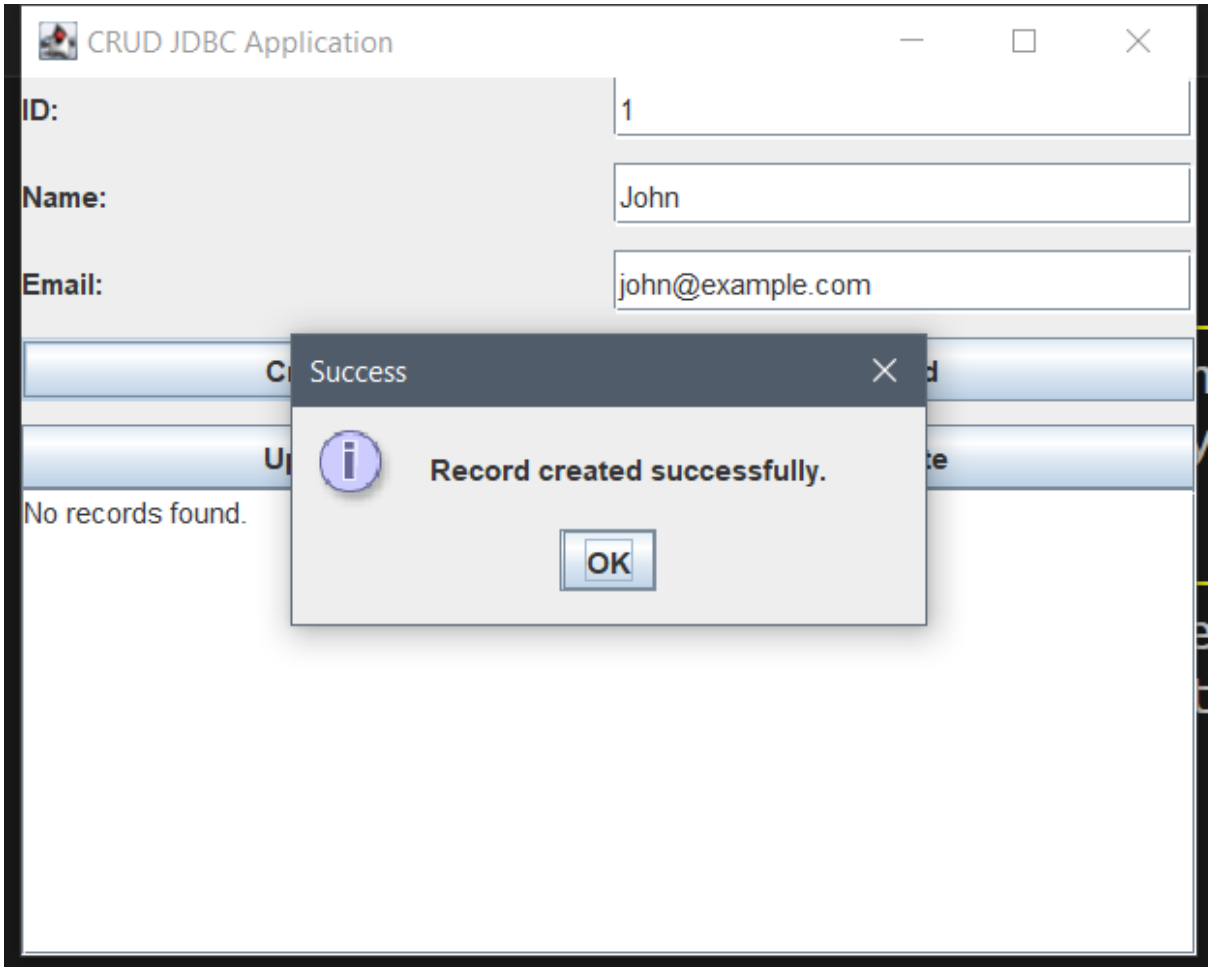
```
      Statement statement = connection.createStatement();
      ResultSet resultSet = statement.executeQuery("SELECT * FROM users");

      int i = 0;
      resultArea.setText("");
      while (resultSet.next()) {
        int id = resultSet.getInt("id");
        String name = resultSet.getString("name");
        String email = resultSet.getString("email");

        resultArea.append("Record #" + (i++) + "\n");
        resultArea.append("ID: " + id + "\n");
        resultArea.append("Name: " + name + "\n");
        resultArea.append("Email: " + email + "\n\n");
      }

      if (i == 0) {
        resultArea.setText("No records found.");
      }
    } catch (SQLException ex) {
      ex.printStackTrace();
      JOptionPane.showMessageDialog(this, "Failed to retrieve records.",
"Error", JOptionPane.ERROR_MESSAGE);
    }
  }

  private void updateRecord() {
    int id = Integer.valueOf(idField.getText());
    String name = nameField.getText();
    String email = emailField.getText();

    try {
```

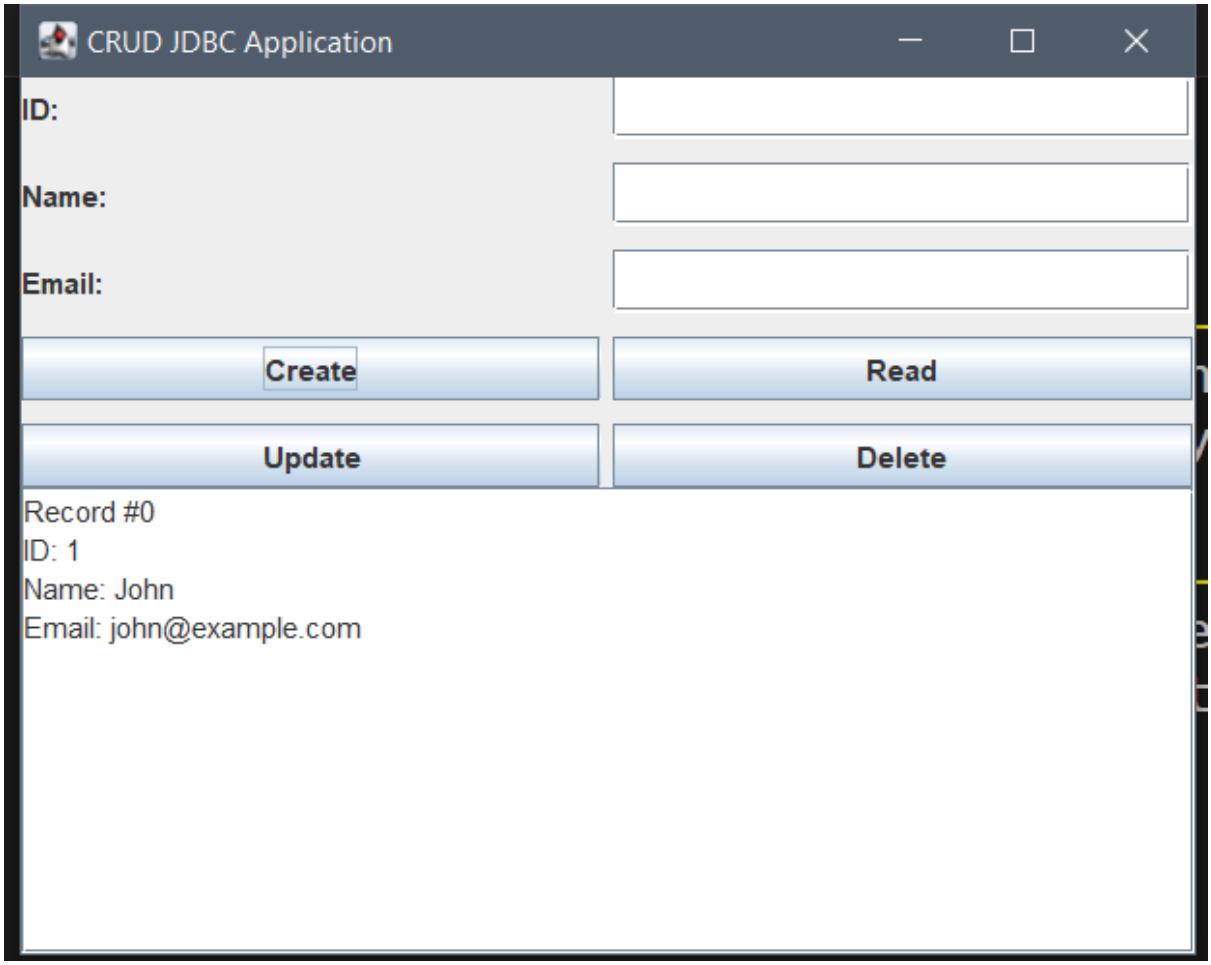
```
      PreparedStatement statement = connection.prepareStatement("UPDATE
users SET name = ?, email = ? WHERE id = ?");

      statement.setString(1, name);

      statement.setString(2, email);

      statement.setInt(3, id);

      int rowsAffected = statement.executeUpdate();


      if (rowsAffected > 0) {

        JOptionPane.showMessageDialog(this, "Record updated
successfully.", "Success", JOptionPane.INFORMATION_MESSAGE);

      } else {

        JOptionPane.showMessageDialog(this, "Record not found.", "Error",
JOptionPane.ERROR_MESSAGE);

      }

    } catch (SQLException ex) {

      ex.printStackTrace();

      JOptionPane.showMessageDialog(this, "Failed to update the record.",
"Error", JOptionPane.ERROR_MESSAGE);

    }


    readRecords();

  }


  private void deleteRecord() {

    int id = Integer.valueOf(idField.getText());


    try {

      PreparedStatement statement = connection.prepareStatement("DELETE
FROM users WHERE id = ?");

      statement.setInt(1, id);

      int rowsAffected = statement.executeUpdate();


      if (rowsAffected > 0) {
```

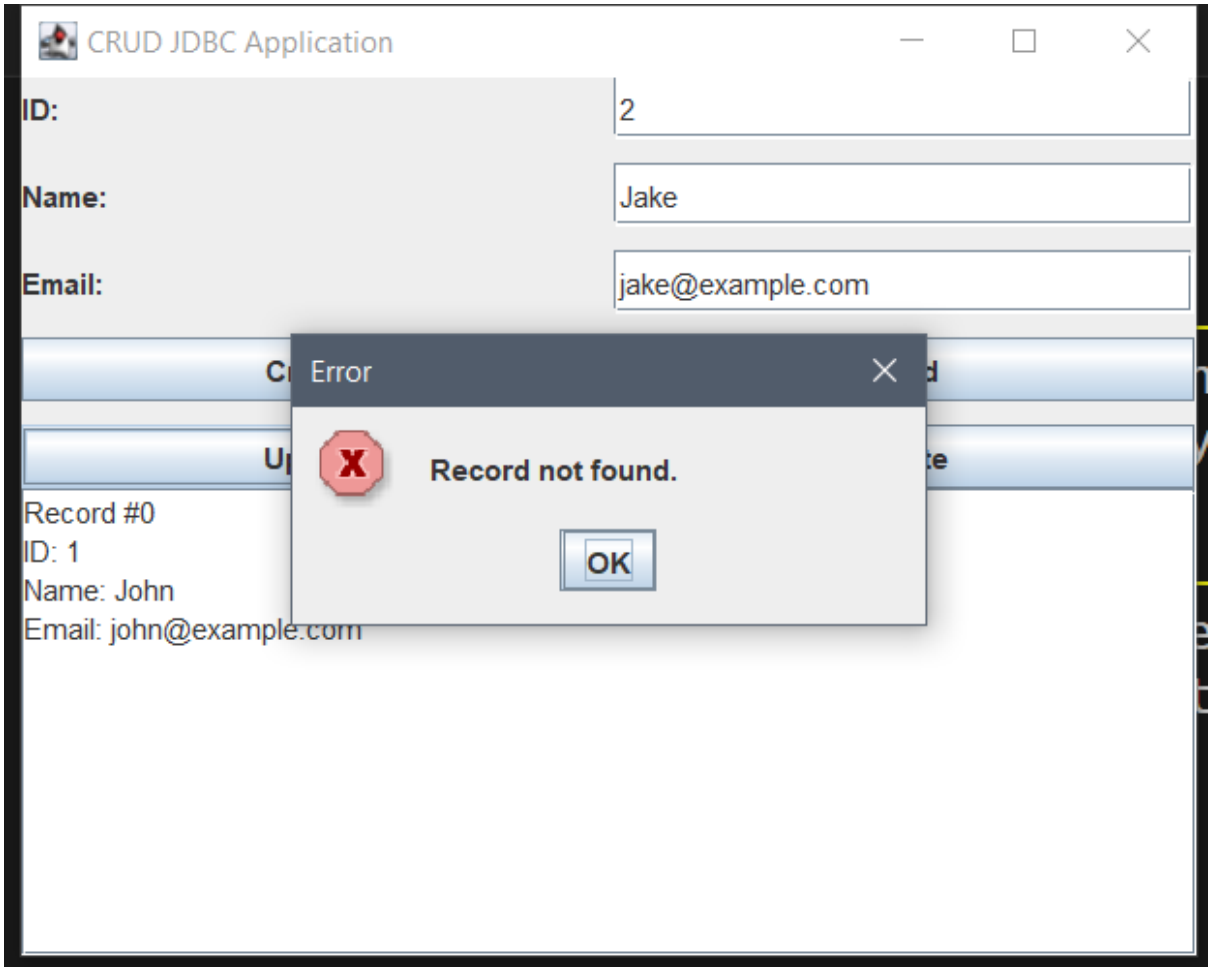
```
        JOptionPane.showMessageDialog(this, "Record deleted
successfully.", "Success", JOptionPane.INFORMATION_MESSAGE);
      } else {
        JOptionPane.showMessageDialog(this, "Record not found.", "Error",
JOptionPane.ERROR_MESSAGE);
      }
    } catch (SQLException ex) {
      ex.printStackTrace();
      JOptionPane.showMessageDialog(this, "Failed to delete the record.",
"Error", JOptionPane.ERROR_MESSAGE);
    }

    readRecords();
  }

  public static void main(String[] args) {
    new CRUDApplication().setVisible(true);
  }
}
```

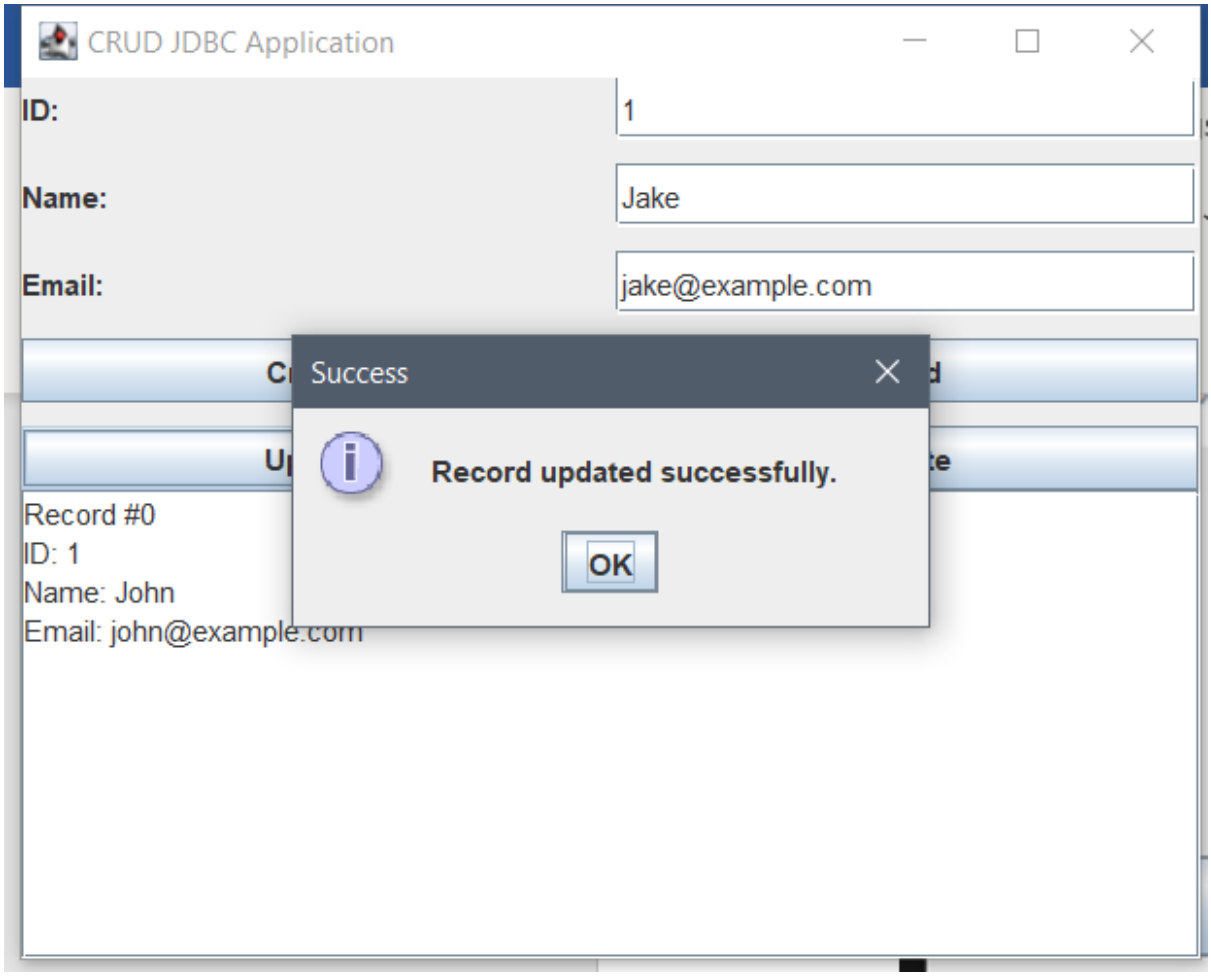
Output

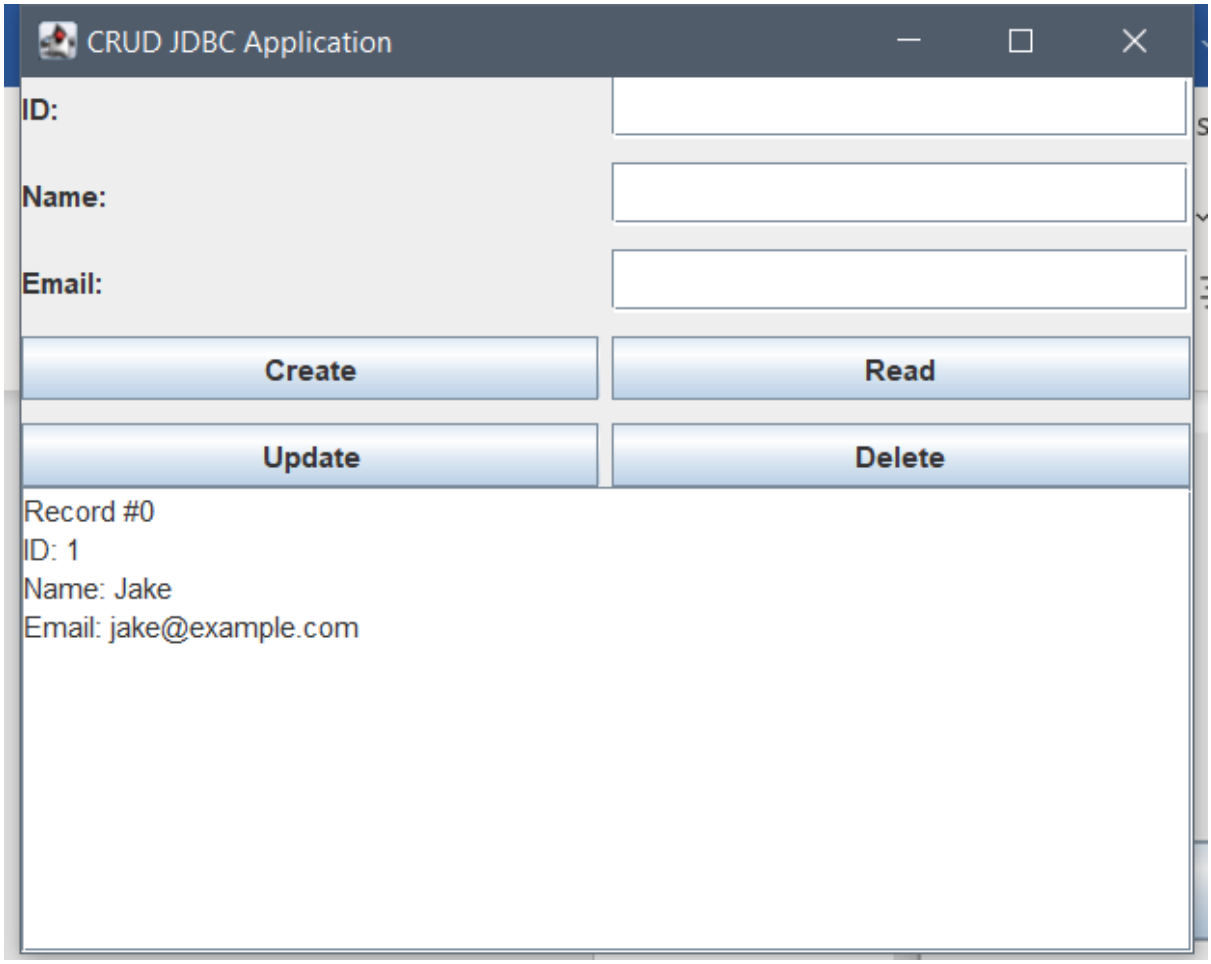

CRUD JDBC Application
ID:
Name:
Email:
Create
Read
Update
Delete
No records found.

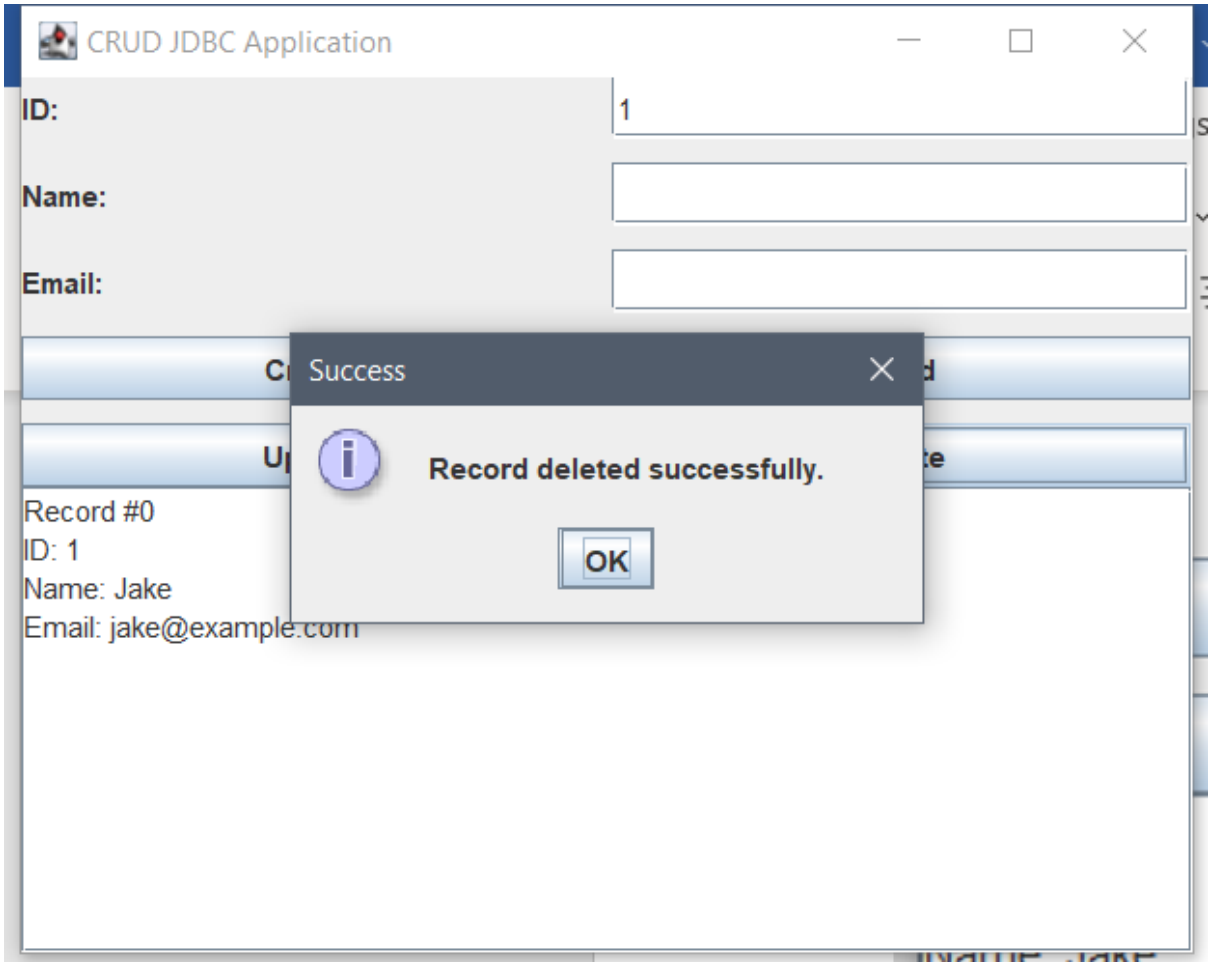

CRUD JDBC Application
ID:
1
Name:
John
Email:
john@example.com
Success
Record created successfully.
OK
No records found.

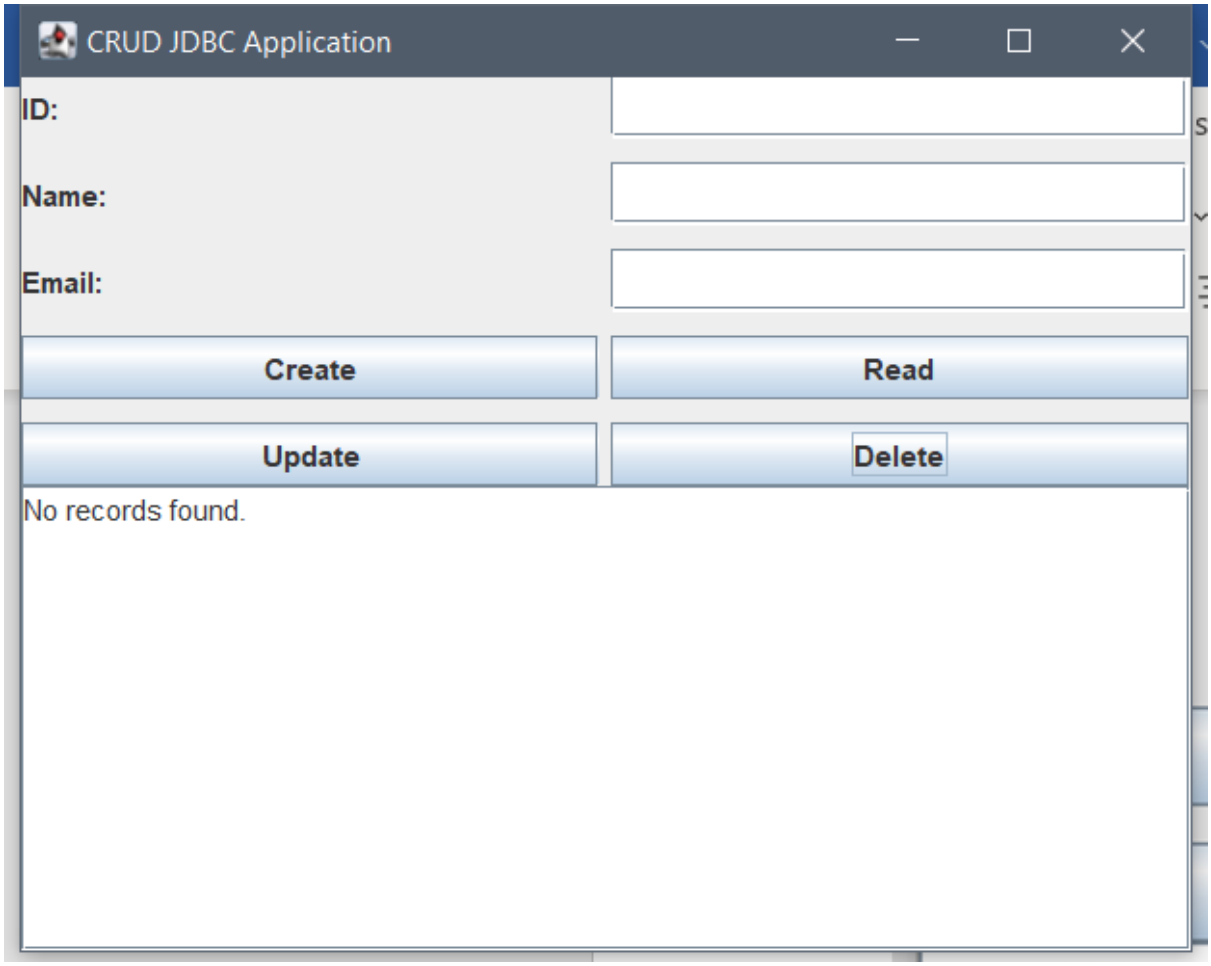

CRUD JDBC Application
ID:
Name:
Email:
Create
Read
Update
Delete
Record #0
ID: 1
Name: John
Email: john@example.com


CRUD JDBC Application
ID:
2
Name:
Jake
Email:
jake@example.com
Error
Record not found.
OK
Record #0
ID: 1
Name: John
Email: john@example.com


CRUD JDBC Application
ID:
1
Name:
Jake
Email:
jake@example.com
Success
Record updated successfully.
OK
Record #0
ID: 1
Name: John
Email: john@example.com


CRUD JDBC Application
ID:
Name:
Email:
Create
Read
Update
Delete
Record #0
ID: 1
Name: Jake
Email: jake@example.com


CRUD JDBC Application
ID:
1
Name:
Email:
Success
Record deleted successfully.
OK
Record #0
ID: 1
Name: Jake
Email: jake@example.com



Q17.

SubmitServlet.java

```
import javax.servlet.*;

import javax.servlet.http.*;

import java.io.*;


public class SubmitServlet extends HttpServlet {

  protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

    String name = request.getParameter("name");

    String email = request.getParameter("email");


    response.setContentType("text/html");

    PrintWriter out = response.getWriter();
```

```
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Form Submission Result</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>Form Submission Result</h1>");
    out.println("<p>Name: " + name + "</p>");
    out.println("<p>Email: " + email + "</p>");
    out.println("</body>");
    out.println("</html>");
  }
}
```

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Form Submission</title>
  </head>
  <body>
    <h1>Form Submission</h1>
    <form action="SubmitServlet" method="post">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name" required /><br /><br />

      <label for="email">Email:</label>
      <input type="email" id="email" name="email" required /><br /><br />

      <input type="submit" value="Submit" />
    </form>
  </body>
</html>
```

Output

Form Submission

Name: Kamran

Email: kamran@example.com

Submit

Form Submission Result

Name: Kamran

Email: kamran@example.com