Banarsidas Chandiwala Institute of Information Technology

Sr. No	Question	Pg. No
1.	Write a Java program to print all odd numbers between 1 to 10	
2.	Write a Java program to find out factorial of a number through recursion	
3.	Write a Java program to accept command line arguments & print them	
4.	Write a Java program to print fibonacci series	
5.	Write a Java program that creates a class accounts with following details: Instance variables: ac_no., name, ac_name, balance Methods: withdrawal (), deposit (), display ().use constructors to initialize members	
6.	Write a Java program to implement constructor overloading	
7.	Write a Java program to count the no. of objects created in a program	
8.	Write a Java program to implement method over ridding & method overloading	
9.	Create a class box having height, width, depth as the instance variables & calculate its volume. Implement constructor overloading in it. Create a subclass named box_new that has weight as an instance variable. Use super in the box_new class to initialize members of the base class	
10.	Write a Java program to implement run time polymorphism	
11.	Write a Java program to implement interface. Create an interface named shape having area () & perimeter () as its methods. Create three classes circle, rectangle & square that implement this interface	
12.	Write a Java program to show multiple inheritance	
13.	Write a Java program to implement vector [use: addelement(),elementat().removeelement(),size().]	
14.	Create a user defined exception named “nomatchexception” that is fired when the string entered by the user is not “india”	
15.	Write a Java program to show even & odd numbers by *thread*	
16.	Write a Java program to iterate through all elements in a array list	

17	Write a Java program to demonstrate the use of equals(), trim() ,length() , substring(), compareto() of string class	
18	Write a Java program to demonstrate the use of equals() and == in Java	
19	Write a Java program to check a word contains the character 'g' in a given string.	
20	Write a Java program on anonymous classes	
21	Write a Java program to highlight the structure/syntax of a lambda expression	
22	Write a Java program in Java to create database table using Java	
23	Write a Java program in Java to insert, update, delete & select records	

Question 1: Write a Java program to print all odd numbers between 1 to 10.

Code:

```
import static java.lang.System.out;
class Program1 {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i += 2) {
            out.println(i);
        }
    }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 1.java"

F:\java lab file>java Program1
1
3
5
7
9
```

Question 2. Write a Java program to find out factorial of a number through recursion.

Code:

```
import static java.lang.System.out;
import java.util.Scanner;
class Program2 {
   static int factorial(int i) {
      if (i == 0)
         return 1;
      else
         return i * factorial(i - 1);
   }

   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      out.print("Enter a number: ");
      int i = input.nextInt();
      out.println("\nFactorial of the number is: " + factorial(i));
   }
}
```

OUTPUT:

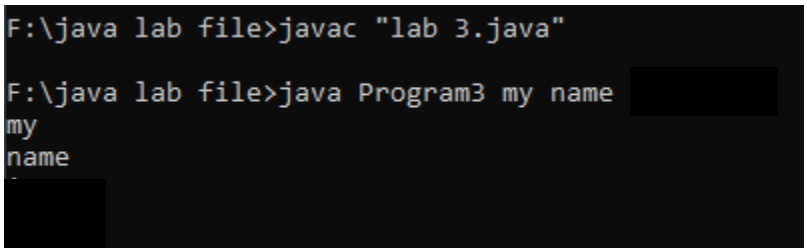
```
F:\java lab file>javac "lab 2.java"

F:\java lab file>java Program2
Enter a number: 5

Factorial of the number is: 120
```

Question 3. Write a Java program to accept command line arguments & print them.

Code

```
import static java.lang.System.out;


class Program3 {
   public static void main(String[] args) {
      for (String s: args) {
         out.println(s);
      }
   }
}
```

OUTPUT:



Question 4. Write a Java program to print fibonacci series.

Code:

```
import static java.lang.System.out;
import java.util.Scanner;

class Program4 {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      out.print("Enter a number: ");
      int n = input.nextInt();
      int a = 0, b = 1, c = a + b;
      out.printf("Fibonnaci series till %d is:\n%d\n%d\n%d\n", n, a, b, c);
      for (int i = 0; i < n - 3; i++) {
         a = b;
         b = c;
         c = a + b;
         out.println(c);
      }
   }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 4.java"

F:\java lab file>java Program4
Enter a number: 8
Fibonnaci series till 8 is:
0
1
1
2
3
5
8
13
```

Question 5. Write a Java program that creates a class accounts with following details:

Code:

Instance variables: ac_no., name, ac_name, balance

Methods: withdrawal(), deposit(), display().use constructors to initialize members.

```
import static java.lang.System.out;

class Accounts {
    private int ac_no;
    private String name, ac_name;
    private double balance;

    public Accounts(int n, String nm, String ac_nm, double bal) {
        ac_no = n;
        name = nm;
        ac_name = ac_nm;
        balance = bal;
        out.println("account created!");
    }

    public void deposit(double amount) {
        balance += amount;
        out.println("amount deposited.\ntotal balance is:" + balance);
    }

    public void withdrawal(double amount) {
        balance -= amount;
        out.println("amount withdrawn.\ntotal balance is:" + balance);
    }

    public void display() {
        out.printf("amount number: %d\nname: %s\naccount name: %s\ntotal balance: %f\n",
```

```
            ac_no, name, ac_name, balance);
    }
}

class Program5 {
    public static void main(String[] args) {
        Accounts ac = new Accounts(501, "John", "54456KLQDH", 10000.0);
        ac.deposit(2000.0);
        ac.withdrawal(3000.0);
        ac.display();
    }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 5.java"

F:\java lab file>java Program5
account created!
amount deposited.
total balance is:12000.0
amount withdrawn.
total balance is:9000.0
amount number: 501
name: John
account name: 54456KLQDH
total balance: 9000.000000
```

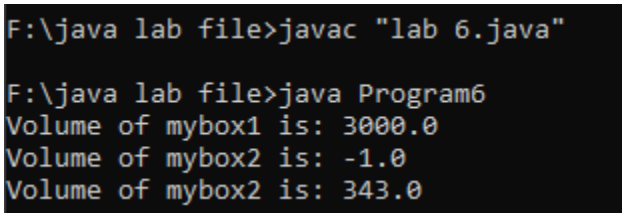
Question 6. Write a Java program to implement constructor overloading.

Code:

```
import static java.lang.System.out;

class Box {
   double width, height, depth;
   Box() {
      width = height = depth = -1;
   }

   Box(double len) {
      width = height = depth = len;
   }

   Box(double w, double h, double d) {
      width = w;
      height = h;
      depth = d;
   }

   double volume() {
      return width * height * depth;
   }
}

class Program6 {
   public static void main(String[] args) {
      Box mybox1 = new Box(10, 20, 15),
         mybox2 = new Box(),
         mybox3 = new Box(7);
         double vol;
```

```
        vol = mybox1.volume();
        out.println("Volume of mybox1 is: " + vol);
        vol = mybox2.volume();
        out.println("Volume of mybox2 is: " + vol);
        vol = mybox3.volume();
        out.println("Volume of mybox2 is: " + vol);
    }
}
```

OUTPUT:



Question 7. Write a Java program to count the no. of objects created in a program.

Code:

```
import static java.lang.System.out;


class Program7 {
   static int objectCount = 0;


   Program7() {
      objectCount++;
   }


   public static void main(String[] args) {
      Program7 p1 = new Program7(),
            p2 = new Program7(),
            p3 = new Program7(),
            p4 = new Program7(),
            p5 = new Program7();


      out.println("Object Created in program is: " + Program7.objectCount);
   }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 7.java"

F:\java lab file>java Program7
Object Created in program is: 5
```

Question 8. Write a Java program to implement method over ridding & method overloading.

Code:

```
import static java.lang.System.out;
import static java.lang.Math.PI;

interface Area {
   void findArea(int i);
}

class SquareArea implements Area {
   @Override
   public void findArea(int side) {
      out.println("Area of square is: " + (side * side));
   }
}

class CircleArea implements Area {
   @Override
   public void findArea(int radius) {
      out.println("Area of circle is: " + (PI * radius * radius));
   }
}

class Program9 {
   static void cube(int value) {
      out.println("Cube of int is: " + (value * value * value));
   }

   static void cube(double value) {
      out.println("Cube of double is: " + (value * value * value));
   }
```

```
public static void main(String[] args) {
        SquareArea sq = new SquareArea();
        sq.findArea(5);
        CircleArea cir = new CircleArea();
        cir.findArea(4);

        out.println();

        cube(5);
        cube(2.5);
    }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 9.java"

F:\java lab file>java Program9
Area of square is: 25
Area of circle is: 50.26548245743669

Cube of int is: 125
Cube of double is: 15.625
```

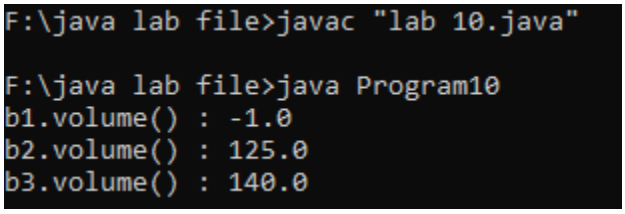
Question 9. Create a class box having height, width, depth as the instance variables & calculate its volume. Implement constructor overloading in it. Create a subclass named box_new that has weight as an instance variable. Use super in the box_new class to initialize members of the base class.

Code:

```
import static java.lang.System.out;
class Box {
   double width, height, depth;
   Box() {
      width = height = depth = -1.0;
   }
   Box(double len) {
      width = height = depth = len;
   }
   Box(double w, double h, double d) {
      width = w;
      height = h;
      depth = d;
   }
   double volume() {
      return width * height * depth;
   }
}

class BoxNew extends Box {
   double weight;
   BoxNew() {
      super();
      weight = 0.0;
   }
   BoxNew(double len, double wt) {
      super(len);
      weight = wt;
```

```
    }    BoxNew(double w, double h, double d, double wt) {
        super(w, h, d);
        weight = wt;
    }
}

class Program10 {
    public static void main(String[] args) {
        BoxNew b1 = new BoxNew(),
               b2 = new BoxNew(5, 30),
               b3 = new BoxNew(4, 7, 5, 12);

        out.println("b1.volume() : " + b1.volume());
        out.println("b2.volume() : " + b2.volume());
        out.println("b3.volume() : " + b3.volume());
    }
}
```

OUTPUT:



Question 10. Write a Java program to implement interface. Create an interface named shape having area () & perimeter () as its methods. Create three classes circle, rectangle & square that implement this interface.

Code:

```
import static java.lang.System.out;
import static java.lang.Math.PI;

interface Shape {
    void area();
    void perimeter();
}

class Circle implements Shape {
    int radius;

    Circle(int r) {
        radius = r;
    }

    @Override
    public void area() {
        out.printf("Area of circle with radius %d is: %f\n", radius, (PI * radius * radius));
    }

    @Override
    public void perimeter() {
        out.printf("Perimeter of circle with radius %d is: %f\n", radius, (2 * PI * radius));
    }
}

class Rectangle implements Shape {
    int length, breadth;
```

```
    Rectangle(int l, int b) {
        length = l;
        breadth = b;
    }

    @Override
    public void area() {
        out.printf("Area of rectangle with length and breadth of %d, %d is: %d\n",
            length, breadth, (length * breadth));
    }

    @Override
    public void perimeter() {
        out.printf("Perimeter of rectangle with length and breadth of %d, %d is: %d\n",
            length, breadth, (2 * (length + breadth)));
    }
}

class Square implements Shape {
    int side;

    Square(int s) {
        side = s;
    }

    @Override
    public void area() {
        out.printf("Area of square with side %d is: %d\n", side, (side * side));
    }
```

```
@Override
    public void perimeter() {
        out.printf("Perimeter of square with side %d is: %d\n", side, (4 * side));
    }
}

class Program12 {
    public static void main(String[] args) {
        Circle c = new Circle(8);
        c.area();
        c.perimeter();

        out.println();

        Rectangle r = new Rectangle(5, 4);
        r.area();
        r.perimeter();

        out.println();

        Square s = new Square(3);
        s.area();
        s.perimeter();
    }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 12.java"

F:\java lab file>java Program12
Area of circle with radius 8 is: 201.061930
Perimeter of circle with radius 8 is: 50.265482

Area of rectangle with length and breadth of 5, 4 is: 20
Perimeter of rectangle with length and breadth of 5, 4 is: 18

Area of square with side 3 is: 9
Perimeter of square with side 3 is: 12
```

Question 11. Write a Java program to implement run time polymorphism.

Code:

```
import static java.lang.System.out;
interface AreaInterface {
    void findArea(int i1, int i2);
}
class Rhombus implements AreaInterface {
    @Override
    public void findArea(int d1, int d2) {
        out.println("Area of rhombus is: " + (0.5 * (d1 * d2)));
    }
}
class Triangle implements AreaInterface {
    @Override
    public void findArea(int base, int height) {
        out.println("Area of triangle is: " + (0.5 * base * height));
    }
}
class Program11 {
    public static void main(String[] args) {
        AreaInterface a = new Rhombus();
        a.findArea(5, 7);
        a = new Triangle();
        a.findArea(4, 5);
    }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 11.java"

F:\java lab file>java Program11
Area of rhombus is: 17.5
Area of triangle is: 10.0
```

Question 12. Write a Java program to show multiple inheritance.

Code:

```
import static java.lang.System.out;

class Student {
   int rollNumber;
   String name;

   void setInfo(int rn, String nm) {
      rollNumber = rn;
      name = nm;
   }

   void printInfo() {
      out.println(name);
      out.println("Roll number: " + rollNumber);
   }
}

class Test extends Student {
   int marks1, marks2;

   Test(int m1, int m2) {
      marks1 = m1;
      marks2 = m2;
   }

   void printMarks() {
      out.println("Marks Obtained:-");
      out.println("MArks 1: " + marks1);
      out.println("MArks 2: " + marks2);
   }
```

```
}

interface Sports {
    int marks = 10;
    void putMarks();
}

class Result extends Test implements Sports {
    int total;

    Result(int m1, int m2) {
        super(m1, m2);
        total = marks1 + marks2 + marks;
    }

    @Override
    public void putMarks() {
        out.println("Sports Marks: " + marks);
    }

    void display() {
        printInfo();
        printMarks();
        putMarks();
        out.println("total: " + total);
    }
}

class Program13 {
    public static void main(String[] args) {
        Result result = new Result(30, 60);
        result.setInfo(5, "John");
```

```
        result.display();
    }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 13.java"

F:\java lab file>java Program13
John
Roll number: 5
Marks Obtained:-
MArks 1: 30
MArks 2: 60
Sports Marks: 10
total: 100
```

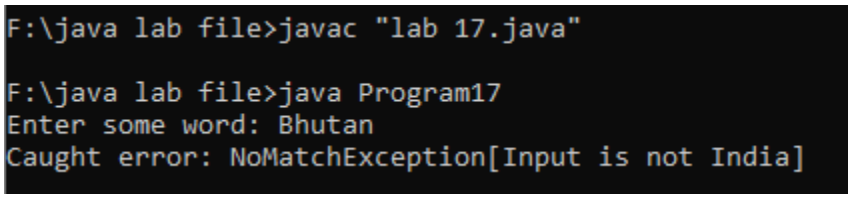
Question 13: Create a user defined exception named “nomatchexception” that is fired when the string entered by the user is not “india”.

Code:

```
import static java.lang.System.out;
import java.util.Scanner;

class NoMatchException extends Exception {
   String detail;

   public NoMatchException(String dtl) {
      detail = dtl;
   }

   @Override
   public String toString() {
      return String.format("NoMatchException[%s]", detail);
   }
}

class Program17 {
   static void checkAndThrow() throws NoMatchException {
      Scanner input = new Scanner(System.in);
      out.print("Enter some word: ");
      String s = input.nextLine();
      if (!(s.equals("india") || s.equals("India") || s.equals("INDIA"))) {
         throw new NoMatchException("Input is not India");
      }
      out.println("Normal Exit.");
   }

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

```
            checkAndThrow();
        } catch (NoMatchException e) {
            out.println("Caught error: " + e);
        }
    }
}
```

OUTPUT:



Question 14: Write a Java program to show even & odd numbers by thread.

import static java.lang.System.out;

Code:

```
class OddNumbersThread extends Thread {
    @Override
    public void run() {
        try {
            for (int i = 1; i <= 10; i += 2) {
                out.println("Odd Number: " + i);
                Thread.sleep(5);
            }
        } catch (InterruptedException e) {
            out.println("Caught error: " + e);
        }
    }
}

class EvenNumbersThread extends Thread {
    @Override
    public void run() {
        try {
            for (int i = 2; i <= 10; i += 2) {
                out.println("Even Number: " + i);
                Thread.sleep(5);
            }
        } catch (InterruptedException e) {
            out.println("Caught error: " + e);
        }
    }
}
```

```
class Program18 {
    public static void main(String[] args) {
        new OddNumbersThread().start();
        new EvenNumbersThread().start();
    }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 18.java"

F:\java lab file>java Program18
Odd Number: 1
Even Number: 2
Odd Number: 3
Even Number: 4
Odd Number: 5
Even Number: 6
Odd Number: 7
Even Number: 8
Odd Number: 9
Even Number: 10
```

Question 15: Write a Java program to implement vector [use: addelement(), elementat(), removeElement(), size().]

Code:

```
import java.util.Scanner; import java.util.Vector;

class Program16 {
public static void main(String[] args) { Vector<String> list = new Vector<String>(); int n = 0;
Scanner scanner = new Scanner(System.in); outerLoop:
while (true) {
System.out.print("\nVector Operations: \n\n1.Add Element\n2.Element At\n3.Remove
At\n4.Size\n5.Exit");
System.out.print("\n\nOption:"); n = scanner.nextInt();
switch (n) {
case 1:



break; case 2:
System.out.print("Enter new Element: "); list.add(scanner.next());


System.out.print("Enter Index: ");


System.out.println(list.elementAt(scanner.nextInt()));
break; case 3:
System.out.print("Enter Index to Remove Element: ");

System.out.println(list.remove(scanner.nextInt())); break;
case 4:
System.out.println("Size="+list.size());
break; case 5:
break outerLoop;
}
}
}
}
```

OUTPUT:

```
Vector Operations:

1.Add Element
2.Element At
3.Remove At
4.Size
5.Exit

Option:1
Enter new Element: 55

Vector Operations:

1.Add Element
2.Element At
3.Remove At
4.Size
5.Exit

Option:2
Enter Index: 0
55

Vector Operations:

1.Add Element
2.Element At
3.Remove At
4.Size
5.Exit

Option:3
Enter Index to Remove Element: 0
55
```

```
Vector Operations:

1.Add Element
2.Element At
3.Remove At
4.Size
5.Exit

Option:4
Size=0

Vector Operations:

1.Add Element
2.Element At
3.Remove At
4.Size
5.Exit

Option:5
```

QUESTION 16: Write a Java program to retrieve an element (at a specified index) from a given array list.

CODE:

```
import java.util.ArrayList;

class Program20 {

	public static void main(String[] args) {
		ArrayList<Integer> arr = new ArrayList<Integer>(4);
		arr.add(15);
		arr.add(25);
		arr.add(35);
		arr.add(45);
		System.out.println("ARRAY LIST: " + arr); int
		element = arr.get(3);
		System.out.println("THE ELEMENT AT INDEX 3: " + element);
	}
}
```

OUTPUT:

```
F:\java lab file>javac "lab 20.java"

F:\java lab file>java Program20
ARRAY LIST: [15, 25, 35, 45]
THE ELEMENT AT INDEX 3: 45
```

QUESTION 17: Write a Java program to demonstrate the use of equals(), trim(), length(), substring(), compareto() of string class.

CODE:

```
class Program24 {
   public static void main(String args[]) {
      String str1="Mohan";
      String str2="Mohan";
      String str3="My Name Is Mohan";
      String str4="       Remove Space";

      System.out.println(str1.equals(str2)); System.out.println("String
      Without Trim: "+str4); System.out.println("Trimmed String: " +
      str4.trim());
      System.out.println("Length of string (" + str3 +"): " + str3.length());
      System.out.println("Substring ("+str3+"): " + str3.substring(4));
      System.out.println(str1.compareTo(str2)); System.out.println(str1.compareTo(str3));

   }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 24.java"

F:\java lab file>java Program24
true
String Without Trim:         Remove Space
Trimmed String: Remove Space
Length of string (My Name Is Mohan):  16
Substring (My Name Is Mohan): ame Is Mohan
0
-10
```

QUESTION 18: Write a Java program to demonstrate the use of equals() and == in Java.

CODE:

```
class Program25 {
   public static void main(String args[]) {
      String str1 ="Hello World";
      String str2 ="Hello World";
      String str3 = new String("Hello World");

      System.out.println(str1.equals(str2));
      System.out.println(str1==str2);
      System.out.println(str1==str3);
      System.out.println(str1.equals(str3));

   }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 25.java"

F:\java lab file>java Program25
true
true
false
true
```

QUESTION 19: Write a Java program to check a word contains the character 'g' in a given string.

CODE:

```
import java.util.Scanner;

class Program42 {

   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      System.out.print("Enter some string: ");
      String s = input.nextLine();
      if (s.contains("g")) {
         System.out.println("String \"" + s + "\" contains character \'g\'");
      } else {
         System.out.println("String \"" + s + "\" does not contains character \'g\'");
      }
   }
}
```

OUTPUT

```
F:\java lab file>javac "lab 42.java"

F:\java lab file>java Program42
Enter some string: good morning
String "good morning" contains character 'g'
```

QUESTION 22: Write a Java program on anonymous classes.

CODE:

```
class Animal {
   public void makeSound() {
      System.out.println("The animal is making some sound.");
   }
}

class Program40 {
   public static void main(String[] args) {
      Animal dog = new Animal() {
         @Override
         public void makeSound() {
            System.out.println("Woof woof!!");
         }
      };

      dog.makeSound();
   }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 40.java"

F:\java lab file>java Program40
Woof woof!!
```

QUESTION 23: Write a Java program to highlight the structure/syntax of a lambda expression.

CODE:

```
    @FunctionalInterface
 interface Operation { int
            calculate(int a);
}

class Program41 {
   public static void main(String[] args) {
      Operation square = (a) -> a * a;
      int num = 5;
      int result = square.calculate(num); System.out.printf("Square
      of %d is: %d", num, result);
   }
}
```

OUTPUT:

```
F:\java lab file>javac "lab 41.java"

F:\java lab file>java Program41
Square of 5 is: 25
F:\java lab file>
```

QUESTION 24: Write a Java program to show database connectivity

CODE:

```
import java.sql.*;

class Program24

{
public static void main(String[] args)
{
      Connection connection = null;
      String query;
      ResultSet result; try
      {
         Class.forName("com.mysql.cj.jdbc.Driver");
         connection = DriverManager.getConnection("jdbc:mysql://localhost:3308/",
                  "12345", "12345");

         Statement stmt = connection.createStatement();
         query = "CREATE DATABASE java_database";
         stmt.executeUpdate(query);
         System.out.println("Database Created Successfully!!");
         connection.close();
      }

      catch (Exception exception)
      {
         System.out.println(exception);
      }

}
}
```

```
Database Created Successfully!!
```

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| java_database      |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

QUESTION 25: Write a Java program in Java to insert, update, delete & select records.

CODE:

```
import java.sql.*;


class Program37


{
   public static void main(String[] args)
   {
      Connection connection = null;
      String query;
      ResultSet result; try
      {
         Class.forName("com.mysql.cj.jdbc.Driver");
         connection =
DriverManager.getConnection("jdbc:mysql://localhost:3308/java_database","12345", "12345");

         Statement stmt = connection.createStatement();

         query = "INSERT INTO Recipies(recipe_id, recipe_name)"
                  + " VALUES"
                  + "(6,\"Momos\"), "
                  + "(7,\"Spring Roll\"), "
                  + "(8,\"Burger\"), "
                  + "(9,\"Pizza\"), "
                  + "(10,\"Garlic Bread\"); ";
          stmt.executeUpdate(query); System.out.println("Inserted
          records into the table...");

          query = "SELECT * FROM Recipies;";
          result = stmt.executeQuery(query);
          while(result.next()){
             System.out.print("Recipie ID: " + result.getInt("recipe_id"));
             System.out.print("Recipie Name: " + result.getString("recipe_name"));
             System.out.println("\n");
         }

         query = "UPDATE Recipies " +
         "SET recipe_name = 'My Recipie' WHERE recipe_id = 4";
         stmt.executeUpdate(query);
         System.out.println("Record Successfully Updated!!!!\n");
         query = "SELECT * FROM Recipies;";
         result = stmt.executeQuery(query);
```

```
            while(result.next()){
                System.out.print("Recipie ID: " + result.getInt("recipe_id"));
               System.out.print("Recipie Name: " + result.getString("recipe_name"));
               System.out.println("\n");
             }


             query = "DELETE FROM Recipies where recipe_id = 3";
             stmt.executeUpdate(query);
             System.out.println("Record Successfully Deleted!!!!\n");
             query = "SELECT * FROM Recipies;";
             result = stmt.executeQuery(query);
             while(result.next()){
                System.out.print("Recipie ID: " + result.getInt("recipe_id"));
                System.out.print("|Recipie Name: " + result.getString("recipe_name"));
                System.out.println("\n");
             }

            connection.close();
        }
        catch (Exception exception)
        {
            System.out.println(exception);
        }

}
}
```

OUTPUT:

```
Inserted records into the table...
Recipie ID: 8Recipie Name: Burger

Recipie ID: 3Recipie Name: Cucumber Salad

Recipie ID: 10Recipie Name: Garlic Bread

Recipie ID: 5Recipie Name: Grilled Cheeze

Recipie ID: 6Recipie Name: Momos

Recipie ID: 4Recipie Name: Pasta

Recipie ID: 9Recipie Name: Pizza

Recipie ID: 7Recipie Name: Spring Roll

Recipie ID: 1Recipie Name: Tacos

Recipie ID: 2Recipie Name: Tomato Soup

Record Successfully Updated!!!!

Recipie ID: 8Recipie Name: Burger

Recipie ID: 3Recipie Name: Cucumber Salad

Recipie ID: 10Recipie Name: Garlic Bread

Recipie ID: 5Recipie Name: Grilled Cheeze

Recipie ID: 6Recipie Name: Momos

Recipie ID: 4Recipie Name: My Recipie

Recipie ID: 9Recipie Name: Pizza

Recipie ID: 7Recipie Name: Spring Roll

Recipie ID: 1Recipie Name: Tacos

Recipie ID: 2Recipie Name: Tomato Soup
```

```
Record Successfully Deleted!!!!

Recipie ID: 8|Recipie Name: Burger

Recipie ID: 10|Recipie Name: Garlic Bread

Recipie ID: 5|Recipie Name: Grilled Cheeze

Recipie ID: 6|Recipie Name: Momos

Recipie ID: 4|Recipie Name: My Recipie

Recipie ID: 9|Recipie Name: Pizza

Recipie ID: 7|Recipie Name: Spring Roll

Recipie ID: 1|Recipie Name: Tacos

Recipie ID: 2|Recipie Name: Tomato Soup
```