Banarsidas Chandiwala Institute Of Information Technology

Java LAB File

Submitted to :-

Mr. Meetender

Submitted by :-

Avinash Ranjan

04011104422

MCA 1st Semester

19.	Write a Java program to check a word contains the character 'g' in a given string.	28.
20.	Write a Java program on anonymous classes .	29.
21.	Write a Java program to highlight the structure/syntax of a lambda Expression.	32.
22.	Write a Java program in Java to create database table using Java.	33.
23.	Write a Java program in Java to insert, update, delete & select records.	34.

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

Code:

```
public class OddNumber {

   public static void main(String[] args) {

      System.out.println("Printing Odd Numbers between 1 and 10");

      for(int i=1; i<=10 ;i++){
      if(i%2 != 0){
         System.out.println(i + " ");
      }
      }
   }
}
```

Output:

```
Printing Odd Numbers between 1 and 10
1
3
5
7
9
BUILD SUCCESSFUL (total time: 1 second)
```

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

Code:

```
import java.util.Scanner;

//program to calculate the factorial of a number using recursion
public class Factorial {
   static int fact(int n){
   int ans  = 1;
      if (n == 1 ){
         return 1;
      }
```

```
      ans = n * fact(n-1);
      return ans;
   }

   public static void main(String[] args) {
      System.out.println("Enter a number for which you want Factorial");
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      System.out.println(fact(n));


   }

}
```

Output:

```
Enter a number for which you want Factorial
6
720
BUILD SUCCESSFUL (total time: 10 seconds)
```

3. Write a java program to accept the command line arguments and print them.

Code:

```
class CommandLine {

   public static void main(String[] args) {

      for (String s: args) {
         System.out.println(s);
      }
   }
}
```

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

Code:

```
import java.util.Scanner;

//program to print Fibonacci Series
public class Fibo {
   static void fib(int n){

      int a =0 , b=1;
      System.out.print( a + " " + b + " ");
      for(int i = 1 ; i <= n ; i++){

        int c = b;
        b = a+ b;
        a  = c;
         System.out.print(b +" ");

      }
   }
   public static void main(String[] args) {
      System.out.println("Enter a number for the elements in fibonacci series : ");
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      fib(n);
   }
}
```

Output:

```
Enter a number for the elements in fibonacci series :
10
0 1 1 2 3 5 8 13 21 34 55 89
BUILD SUCCESSFUL (total time: 6 seconds)
```

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.

Code:

```
public class Accounts {
   private long 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;
      System.out.println("Account created!");
      this.display();
   }
   public void deposit(double amount) {
      balance += amount;
      System.out.println("Amount deposited :"+ amount+ "\ntotal balance is:" + balance);
   }

   public void withdrawal(double amount) {
      if( balance >= amount)
      {
       balance -= amount;
       System.out.println("Amount withdrawn :"+ amount+ "\nTotal balance is:" + balance);
      }
      else{
         System.err.println("Insuffecient Balance! Withdrawls amount :"+ amount+ "\nTotal
balance is:" + balance);
      }

   }

   public void display() {
      System.out.println("Account Number : " + ac_no);
```

```
    System.out.println("Account Name : " + ac_name);
    System.out.println("Holder Name : " + name);
    System.out.println("Balance : " + balance);
  }

  public static void main(String[] args) {
    Accounts account1 = new Accounts(335567,"Ram","Savings",10000);
    account1.withdrawal(5000);
    account1.deposit(2000);
    account1.withdrawal(8000);
    account1.display();
  }

  }
```

Output:

```
Account created!
Account Number : 335567
Account Name : Savings
Holder Name : Ram
Balance : 10000.0
Amount withdrawn :5000.0
Total balance is:5000.0
Amount deposited :2000.0
total balance is:7000.0
Insuffecient Balance! Withdrawls amount :8000.0 Total balance is:7000.0
Account Number : 335567
Account Name : Savings
Holder Name : Ram
Balance : 7000.0
BUILD SUCCESSFUL (total time: 1 second)
```

6. Write a Java program to implement constructor overloading.

Code:

```
class Student{
   int rollNo;
   String name ;

   Student(int roll , String n){
      rollNo = roll;
      name  = n;
       System.out.println("New Student Added with name : " + n + " Roll No :" + roll);
   }

   Student(int roll){
     rollNo = roll;
      System.out.println("New Student Added with Roll No :" + roll);
   }
   Student(){
      System.out.println("New Student Added");

   }
}

public class ConstructorOverloading {
   public static void main(String[] args) {
      Student s1 = new Student();
      Student s2 = new Student(101);
      Student s3 = new Student(101,"Ram");
   }
```

Output:

```
New Student Added
New Student Added with Roll No :101
New Student Added with name : Ram Roll No :101
BUILD SUCCESSFUL (total time: 1 second)
```

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

Code:

```
class Student{
   int rollNo;
   String name ;
   static int students = 0;

   Student(int roll , String n){
     rollNo = roll;
     name  = n;
     students ++;
   }

   static int getTotalStudents(){
   return students;
   }
}
public class ObjectCount {

      public static void main(String[] args) {
         Student s1  = new Student(101,"Tony");
         Student s2  = new Student(102,"Richard");
         Student s3  = new Student(103,"Peter");

 System.out.println("Total no of objects of Student class = "+ Student.getTotalStudents());

      }

   }
```

Output:

```
Total no of objects of Student class = 3
BUILD SUCCESSFUL (total time: 1 second)
```

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

Code:

```
//program to show Overloading and Overriding

class Rectangle{

   float length, breadth;

   public float getArea(){
   return length * breadth;
   }

}


class Square extends Rectangle{

   float side;

   Square (float s){
   side  = s;
   }
   @Override
   public float getArea(){
   return side * side ;
   }

   public  void display(){ //overloading display
      System.out.println("This is a square object");
   }

  public void display(float s){ //overloading display
    System.out.println("This is a square object with side : "+ s);
   }

}

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

```
        Square s1 = new Square(5);
        s1.display();
        s1.display(s1.side);
        System.out.println("Area : " +s1.getArea());

    }

}
```

Output:

```
This is a square object
This is a square object with side : 5.0
Area : 25.0
BUILD SUCCESSFUL (total time: 1 second)
```

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:

```
public class Box {
  double height, width, depth;

  Box(double h, double w, double d) {
    height = h;
    width = w;
    depth = d;
     System.out.println("Box Created with \n Height :" + height + "\n Width :" + width + "\n
Depth :"+ depth);
  }

  Box(double side) {
    height = width = depth = side;
     System.out.println("Box Created with Sides :" + side);
  }

  double volume() {
    return height * width * depth;
  }
}
class BoxNew extends Box {
  double weight;

  BoxNew(double h, double w, double d, double weight) {
    super(h, w, d);
    this.weight = weight;
     System.out.println(" Weight:"+ weight);
  }
}

class Main{
   public static void main(String[] args) {
      Box b1 = new Box(3,4,4);
       Box b2 = new Box(5);
```

```
        System.out.println("Volume Of Box :" + b1.volume());
        BoxNew bn = new BoxNew(2, 3, 3, 7);

    }
}
```

10. Write a Java program to implement run time polymorphism.

Code:

```
class Shape {
  void draw() {
    System.out.println("Drawing Shape");
  }
}

class Circle extends Shape {
  @Override
  void draw() {
    System.out.println("Drawing Circle");
  }
}

class Square extends Shape {
    @Override
  void draw() {
    System.out.println("Drawing Square");
  }
}
public class RuntimePolymorphism {
     public static void main(String[] args) {
    Shape s;
    s = new Circle();
    s.draw();
    s = new Square();
    s.draw();
  }

}
```

Output:

```
Drawing Circle
Drawing Square
BUILD SUCCESSFUL (total time: 1 second)
```

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.

Code:

```
interface Shape {
  double area();
  double perimeter();
}

class Circle implements Shape {
   double radius;

  final double PI = 3.147;

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

  @Override
  public double area() {
    return PI * radius * radius;
  }

  @Override
  public double perimeter() {
    return 2 * PI * radius;
  }
}

class Rectangle implements Shape {
   double length, width;

   Rectangle(double length, double width) {
    this.length = length;
    this.width = width;
  }
```

```
 @Override
 public double area() {
  return length * width;
 }

 @Override
 public double perimeter() {
  return 2 * (length + width);
 }
}

class Square implements Shape {
  double side;

 Square(double side) {
  this.side = side;
 }

 @Override
 public double area() {
  return side * side;
 }

 @Override
 public double perimeter() {
  return 4 * side;
 }
}
 class ImplementingInterface {
  public static void main(String[] args) {
     Rectangle rect = new Rectangle(4, 6);
     Square square = new Square(5);
     Circle circle = new Circle(4);

     System.out.println("Area of Rectangle" + rect.area());
     System.out.println("Perimeter of Rectangle" + rect.perimeter());

     System.out.println("Area of Square" + square.area());
     System.out.println("Perimeter of Square" + square.perimeter());
```

```
        System.out.println("Area of Circle" + circle.area());
        System.out.println("Perimeter of Circle" + circle.perimeter());
    }

}
```

Output:

```
Area of Rectangle24.0
Perimeter of Rectangle20.0
Area of Square25.0
Perimeter of Square20.0
Area of Circle50.352
Perimeter of Circle25.176
BUILD SUCCESSFUL (total time: 0 seconds)
```

12. Write a Java program to show multiple inheritance.

Code:

```
//program to use multiple inheritance using interfaces

interface area{
float getArea();
}

interface perimeter{
float getPerimeter();
}
class Square implements area, perimeter{

   float side;

   Square(float s){side = s;}

   @Override
   public float getArea(){
   return side*side;
   }

   @Override
   public float getPerimeter(){
   return side * 4;
   }

}

//not compulsary for the program , using lambda expressions
class Rectangle{

   float length, breadth;

    area a = ()->  length* breadth;
    perimeter p = ()-> (length + breadth) * 2;

}
```

```
public class MultipleInheritance {
   public static void main(String[] args) {
      Square s1 = new Square(5);
       System.out.println("Area : " +s1.getArea());
       System.out.println("Perimeter : " +  s1.getPerimeter());
   }

}
```

Output:

```
Area : 25.0
Perimeter : 20.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

Code:

```
import java.util.Scanner;

class NoMatchFoundException extends Exception
{
@Override
   public String toString(){
   return "NoMatchFoundException [The input is Not India]";
   }
}

public class NewException  {
   public static void main(String[] args) {
      String s;
      System.out.println("Enter a Sring");
      Scanner  sc = new Scanner(System.in);
      s= sc.nextLine();

      if(! s.equalsIgnoreCase("India"))
      {
         try {
            throw new NoMatchFoundException();
         } catch (NoMatchFoundException ex) {
            System.err.println(ex);
         }
      }
   }
```

Output:

```
Enter a Sring
Australia
NoMatchFoundException [The input is Not India]
BUILD SUCCESSFUL (total time: 11 seconds)
```

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

Code:

```
class EvenThread extends Thread{

  @Override
   public void run(){
   for(int i = 1; i<= 10 ;i++)
   {
      if(i % 2 == 0)
      {
         try {
            System.out.println("Even Number : " + i);
            Thread.sleep(5);
         } catch (InterruptedException ex) {
            System.err.println(ex);
         }
      }
   }
   }
}

class OddThread extends Thread{

  @Override
   public void run(){
   for(int i = 1; i<= 10 ;i++)
   {
      if(i % 2 != 0)
      {
         try {
            System.out.println("Odd Number : " + i);
            Thread.sleep(5);
         } catch (InterruptedException ex) {
            System.err.println(ex);
         }
      }
   }
```

```
  }
}
public class EvenOddThread {

  public static void main(String[] args) {
    EvenThread t1= new EvenThread();
    OddThread t2 = new OddThread();
    t1.start();
    t2.start();
  }


}
```

Output:

```
Odd Number : 1
Even Number : 2
Odd Number : 3
Even Number : 4
Even Number : 6
Odd Number : 5
Odd Number : 7
Even Number : 8
Odd Number : 9
Even Number : 10
BUILD SUCCESSFUL (total time: 1 second)
```

15. Write a Java program to implement vector

[use: addelement(),elementat().removeelement(),size().]

Code:

```
import java.util.Vector;

public class VectorImplementation {
  public static void main(String[] args) {
    Vector<Integer> v = new Vector<>();

    // Adding elements to the vector
    v.addElement(10);
    v.addElement(20);
    v.addElement(30);
    v.addElement(40);

    System.out.println("Vector elements: " + v);

    // Accessing element at a particular index
    System.out.println("Element at index 2: " + v.elementAt(2));

    // Removing an element from the vector
    v.removeElement(30);
    System.out.println("Vector elements after removal: " + v);

    // Checking the size of the vector
    System.out.println("Size of the vector: " + v.size());
  }
}
```

Output:

```
Vector elements: [10, 20, 30, 40]
Element at index 2: 30
Vector elements after removal: [10, 20, 40]
Size of the vector: 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

16.Write a Java program to iterate through all elements in a array list , and retrieve an element (at a specified index)

Code:

```
import java.util.ArrayList;
import java.util.Iterator;

public class ArrayListIteration {
  public static void main(String[] args) {
    ArrayList<String> list = new ArrayList<>();

    // Adding elements to the ArrayList
    list.add("apple");
    list.add("banana");
    list.add("cherry");
    list.add("dates");

    System.out.println("ArrayList elements: " + list);

    // Iterating through the ArrayList using for-each loop
    System.out.println("Iterating through ArrayList using for-each loop:");
    for (String item : list) {
      System.out.println(item);
    }

    // Iterating through the ArrayList using enhanced for loop
    System.out.println("\nIterating through ArrayList using enhanced for loop:");
    for (int i = 0; i < list.size(); i++) {
      System.out.println(list.get(i));
    }

    // Iterating through the ArrayList using Iterator
    System.out.println("\nIterating through ArrayList using Iterator:");
    Iterator<String> iterator = list.iterator();
    while (iterator.hasNext()) {
      System.out.println(iterator.next());
    }

    //accessing element at a given position.
      System.out.println("Item at Index 0 : " + list.get(0));
      System.out.println("Item at Index 0 : " + list.get(3));
  }
}
```

Output:

```
ArrayList elements: [apple, banana, cherry, dates]
Iterating through ArrayList using for-each loop:
apple
banana
cherry
dates

Iterating through ArrayList using enhanced for loop:
apple
banana
cherry
dates

Iterating through ArrayList using Iterator:
apple
banana
cherry
dates
Item at Index 0 : apple
Item at Index 0 : dates
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

```
public class StringMethods {
  public static void main(String[] args) {
    String str1 = "Hello World";
    String str2 = "    Hello World    ";
    String str3 = "Hello";
    String str4 = "WORLD";

    // Demonstrating the use of equals() method
    System.out.println("Using equals() method: ");
    System.out.println(str1.equals(str2));
    System.out.println(str1.equalsIgnoreCase(str4));

    // Demonstrating the use of trim() method
    System.out.println("\nUsing trim() method: ");
    System.out.println(str2.trim());

    //Using length() method
    System.out.println("\nUsing length() method: ");
    System.out.println(str1.length());

    // using substring() method
    System.out.println("\nUsing substring() method: ");
    System.out.println(str1.substring(6));
    System.out.println(str1.substring(0, 5));

    //using compareTo() method
    System.out.println("\nUsing compareTo() method: ");
    System.out.println(str1.compareTo(str3));
    System.out.println(str3.compareTo(str2));

  }
}
```

Output:

```
Using equals() method:
false
false

Using trim() method:
Hello World

Using length() method:
11

Using substring() method:
World
Hello

Using compareTo() method:
6
40
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

Code:

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

    // using == operator
    System.out.println("Using == operator: ");
    System.out.println(str1 == str2);
    System.out.println(str1 == str3);

    // using equals() method
    System.out.println("\nUsing equals() method: ");
    System.out.println(str1.equals(str2));
    System.out.println(str1.equals(str3));
  }
}
```

Output:

```
Using == operator:
false
true

Using equals() method:
true
true
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

Code:

```
public class WordContainsChar {
  public static void main(String[] args) {
    String str = "This is a sample string.";


     System.out.println("String : " + str);
     System.out.println("Contains 'g' : "+str.contains("g"));

  }
}
```

Output:

```
String : This is a sample string.
Contains 'g' : true
BUILD SUCCESSFUL (total time: 0 seconds)
```

20. Write a java program to create a folder named "java.docx" in which you have to make a file as abc.java and other 3 text files as,total_char, total_lines and total_words. Read the data from the abc.java file and write the values to the 3 files respectively as per the name . eg: write the total no of words in the abc.java file in total_words.txt.

Code:

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;

public class FileHandlingProgram {

   public static void main(String[] args) {

      File dir = new File ("C://java.docx");
      File file = new File(dir , "abc.java");
      File total_chars = new File(dir , "total_char.txt");
      File total_words = new File(dir , "total_words.txt");
      File total_lines = new File(dir , "total_lines.txt");

         try{
         // setting up the directory

         if(! dir.exists()){
         dir.mkdir();
            System.out.println("java.docx Folder created");
         }
         else{
            System.out.println("java.docx Folder Already Exist");
         }


         // setting up the files

         if(! file.exists()){
         file.createNewFile();
            System.out.println("temp.txt created");
         }
         else{
            System.out.println("temp.txt already exist");
         }

         total_words.createNewFile();
```

```
total_chars.createNewFile();
total_lines.createNewFile();



//Setting Up Streams
FileInputStream fileReader = new FileInputStream(file);
FileOutputStream fileWriter  = new FileOutputStream(file);
FileOutputStream  writeCharCount = new FileOutputStream(total_chars);
FileOutputStream  writeLineCount = new FileOutputStream(total_lines);
FileOutputStream  writeWordCount = new FileOutputStream(total_words);

String str =   "class Test{\n" +
            "\n" +
            "      public static void main(String[] args) {\n" +
            "          System.out.println(\"Bye Mars!\");\n" +
            "       }\n" +
            "}";


//writing in the file abc.java
fileWriter.write(str.getBytes());


//reading from the file
String code = new String( fileReader.readAllBytes());

//getting the lines using '\n' escape character
Integer l = code.split("\n").length;
String lines = l.toString();

//getting the words using '\\s+' escape character s+ for ignoring exta spaces
Integer w = code.split("\\s+").length;
String words = w.toString();

Integer c = 0;
//iterating through each word and counting the characters in  them.
//incrementing the character count through each word.
for(String s: code.split("\\s+")){
   c += s.length();
}
String chars = c.toString();

//writing to values of lines, words and characters to the files.
   writeCharCount.write(chars.getBytes());
   writeLineCount.write(lines.getBytes());
```

```
            writeWordCount.write(words.getBytes());
            System.out.println("Data Inserted in the files");

            //closing the strams
            fileReader.close();
            fileWriter.close();
            writeCharCount.close();
            writeLineCount.close();
            writeWordCount.close();




        }

        catch(Exception e){
           System.err.println(e);
        }
    }
}
```

Output:

```
java.docx Folder created
temp.txt created
Data Inserted in the files
BUILD SUCCESSFUL (total time: 0 seconds)
```

21. Write a java program to the website details like its ip address,port number ,protocols etc

Code:

```
import java.net.URL;
import java.net.InetAddress;
public class WebsiteInfo {
   public static void main(String[] args) {
      try {
         URL url = new URL("http://www.google.com/");
         InetAddress in = InetAddress.getByName("www.google.com");




         System.out.println("Port :" + url.getPort());
         System.out.println("Protocol :" + url.getProtocol());
         System.out.println("Host :" + url.getHost());
         System.out.println("Address :" + in.getHostAddress());

      } catch (Exception ex) {
         System.err.println(ex);
      }

   }

}
```

Output:

```
Port :-1
Protocol :http
Host :www.google.com
Address :142.250.193.4
BUILD SUCCESSFUL (total time: 1 second)
```

22. write a java program to implement anonymous class

Code:

```
interface calc{
 int sum(int a, int b);
}

public class AnonymousClass {

    public static void main(String[] args) {
        //using anonymous class
     calc c = new calc(){
                    @Override
                    public int sum(int a,int b){
                    System.out.println("Implementing Anonymous Class");
                    return a + b;
                            }
                         };
        System.out.println( c.sum(7, 3));
    }

}
```

Output:

```
Implementing Anonymous Class
10
BUILD SUCCESSFUL (total time: 1 second)
```

23.Write a java program to implement lambda expression.

Code:

```
interface greet{
void msg();
}
interface draw{
    void drawing(String s);
}
public class LambdaExpression {

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

```
        //with no args
          greet g = () -> {System.out.println("Bye Mars!");};
          g.msg();

          //withs args
          draw d = (shape) -> System.out.println("Drawing " + shape);
          d.drawing("circle");

    }
```

Output:

```
Bye Mars!
Drawing circle
BUILD SUCCESSFUL (total time: 1 second)
```

24. Write a java program to show database connectivity.

Code:

```
import java.sql.Connection;
import java.sql.DriverManager;

public class DatabaseConnection {
    public static Connection  connect(){
      Connection  conn =null;
        try
        {
         Class.forName("oracle.jdbc.driver.OracleDriver");
            conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "admin");
        }
        catch(Exception e){
            System.out.println("Exception in DatabaseConnection class connect() method");
        }
    return conn;
    }
    public static void main(String[] args) {

     Connection conn = DatabaseConnection.connect();
     if(conn != null){
        System.out.println("Connection Successful");
     }
     else{
        System.out.println("Connection Failed");
     }

    }
```

Output:

```
Connection Successful
BUILD SUCCESSFUL (total time: 1 second)
```

25.Write a java program to perform basic sql commands such as create, insert, update, delete.

Code:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.ResultSet;

public class DatabaseConnection {
    public static Connection  connect(){
      Connection  conn =null;
        try
        {
         Class.forName("oracle.jdbc.driver.OracleDriver");
            conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "admin");
        }
        catch(Exception e){
            System.out.println("Exception in DatabaseConnection class connect() method");
        }
    return conn;
    }

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

        Connection conn = DatabaseConnection.connect();

        try{
         Statement st = conn.createStatement();

         //creating a table
        String createQuery = "create table users (userId number(5) ,uname varchar2(20))";
        st.executeUpdate(createQuery);
        System.out.println("table Created");

     //inserting value in table using Insert

     String[] insertValues = {  "Insert into users values(104,'Ram')",
                                "Insert into users values(102,'Krishna')",
                                "Insert into users values(103,'Vishnu')"};
     for(String data : insertValues)  {

     if(st.executeUpdate(data) > 0){
             System.out.println("Inserted");
         }else{
            System.out.println("Not Inserted");
         }

     }

        //update query
        String updateQuery  ="Update users set uname = 'Richard' where userId = 104";
         if(st.executeUpdate(updateQuery) > 0){
             System.out.println("Value updated");
```

```
        }else{
           System.out.println("Not updated");
        }
       //delete query
       String deleteQuery  ="Delete from users where userId = 104";
        if(st.executeUpdate(deleteQuery) > 0){
            System.out.println("Value Deleted");
        }else{
           System.out.println("Not Deleted");
        }

       //Fetching data from the table using Select
        String query = "Select * from users";
        ResultSet rs = st.executeQuery(query);
        while(rs.next()){
            System.out.print( "Id :"+ rs.getString(1) + "   ");
            System.out.print("Name : " + rs.getString(2));
            System.out.println("");

        }
        conn.close();

    }

       catch(Exception e)
       {
           System.err.println(e);
       }

      }

   }
```

Output:

```
table Created
Inserted
Inserted
Inserted
Value updated
Value Deleted
Id :102   Name : Krishna
Id :103   Name : Vishnu
BUILD SUCCESSFUL (total time: 0 seconds)
```