# CSL 203 OPERATING SYSTEMS LAB

LABORATORY RECORD

# SUBMITTED BY

## << >>

**<< >>**

## << >>

**to**

the APJ Abdul Kalam Technological University

*Third Semester B. Tech Degree Lab Examination (2019 Scheme)*



**Department of Computer Science and Engineering**

# N.S.S. College of Engineering, Palakkad February 2022

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**N.S.S. COLLEGE OF ENGINEERING**

***Name*** …………………………………………………………………………………………………….

***Class*** …………………………………………………………. ***Roll No*** ………………………………..

Certified that this is the bonafide record of the work done by the above student in the

**Subject:**

**Subject Code:**

Staff-in-charge Head of the Department

*Submitted for practical examination held on ..*…………………………………………………………

**Registration No:**

Internal Examiner External Examiner

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Contents*** | | | | |
| ***Exp. No.*** | ***Date*** | ***Name of Experiment*** | ***Page No.*** | ***Signature*** |
| 1 | 1-12-21 | String is a palindrome or not | 5 |  |
| 2 | 1-12-21 | Frequency of characters in a string | 7 |  |
| 3 | 1-12-21 | Matrix multiplication | 9 |  |
| 4 | 8-12-21 | Inheritance | 14 |  |
| 5 | 15-12-21 | Polymorphism | 18 |  |
| 6 | 5-01-22 | Garbage Collector | 20 |  |
| 7 | 19-01-22 | File Handling with reader/writer | 21 |  |
| 8 | 19-01-22 | File copy along with file related exceptions | 23 |  |
| 9 | 19-01-22 | Usage of try, catch, throws and finally | 28 |  |
| 10 | 02-02-22 | Multi-threaded program | 30 |  |
| 11 | 02-02-22 | Thread Synchronization | 33 |  |
| 12 | 09-02-22 | String Tokenizer | 36 |  |
| 13 | 09-02-22 | Doubly Linked List | 38 |  |
| 14 | 16-02-22 | Simple Calculator using Java Swing | 40 |  |
| 15 | 16-02-22 | Simulate a traffic light | 45 |  |
| 16 | 16-02-22 | Display all records from a table using JDBC | 48 |  |

**Exp. No: 1 Date:01/12/2021**

**Aim**

Write a Java program that checks whether a given string is a palindrome or not.

**Algorithm**

Step 1: Start

Step 2: Input a text

Step 3: reverse the text

Step 4: compare both the texts

Step 5: if both are same, print “its a palindrome”

Else

print “not a palindrome”

Step 6: Stop

**Source Code**

import java.util.\*;

public class palindrome

{

public static void main(String args[])

{

String a, b = "";

Scanner s = new Scanner(System.in);

System.out.print("Enter the string you want to check:");

a = s.nextLine();

int n = a.length();

for(int i = n - 1; i >= 0; i--)

{

b = b + a.charAt(i);

}

if(a.equalsIgnoreCase(b))

{

System.out.println("The string is palindrome.");

}

else

{

System.out.println("The string is not palindrome.");

}

}

}

**Output**

Enter the string you want to check: Malayalam

The string is a palindrome.

Enter the string you want to check: bobby

The string is not a palindrome.

Enter the string you want to check: deed

The string is a palindrome.

**Result**

The following code has been executed successfully

**Exp. No: 2 Date:01/12/2021**

**Aim**

Write a Java Program to find the frequency of a given character in a string.

**Algorithm**

Step 1: Start

Step 2: Initialize frequency=0

Step 3: Enter a string.

Step 4: Enter key character to be checked.

Step 5: Compare each letter in string with key char.

Step 6: increament frequency with each true time

Step 7: Stop

**Source Code**

import java.util.\*;

public class frequency

{

public static void main(String args[])

{

String a;

char ch;

int i,frequency=0;

Scanner s = new Scanner(System.in);

System.out.print("Enter the string:");

a = s.nextLine();

System.out.print("Enter the character to be searched:");

ch = s.nextLine().charAt(0);

for(i=0;i<a.length();i++)

{

if(ch == a.charAt(i))

{

frequency++;

}

}

System.out.print("Frequency of " + ch +" = " + frequency);

s.close();

}

}

**Output**

Enter the string: malayalam

Enter the character to be searched: a

Frequency of a = 4

Enter the string: bobby

Enter the character to be searched: b

Frequency of b = 3

Enter the string: book

Enter the character to be searched: 2

Frequency of o = 2

**Result**

The following code has been executed successfully

**Exp. No: 3 Date:1/12/2021**

**Aim**

Write a Java program to multiply two given matrices

**Algorithm**

Step 1: Start

Step 2: Declare variables and initialize necessary variables

Step 3: Enter the element of matrices by row wise using loops

Step 4: Check the number of rows and column of first and second matrices

Step 5: If the number of rows of the first matrix is equal to the number of columns of the

second matrix, go to step 6.

Else

print matrix multiplication is not possible.

Step 6: Multiply the matrices using nested loops.

Step 7: Print the product in matrix form as console output.

Step 8: Stop

**Source Code**

import java.util.Scanner;

public class matrixmulti

{

public static void main(String args[])

{

int r1, r2,c1,c2,i,j,k,sum;

Scanner in = new Scanner(System.in);

System.out.println("Enter the number of rows of the first matrix");

r1 = in.nextInt();

System.out.println("Enter the number columns of the first matrix");

c1 = in.nextInt();

System.out.println("Enter the number of rows of the second matrix");

r2 = in.nextInt();

System.out.println("Enter the number of columns of the second matrix");

c2 = in.nextInt();

if(c1==r2)

{

int mat1[][] = new int[r1][c1];

int mat2[][] = new int[r2][c2];

int res[][] = new int[r1][c2];

System.out.println("Enter the elements of the first matrix");

for ( i= 0 ; i < r1 ; i++ )

{

for ( j= 0 ; j < c1 ;j++ )

{

mat1[i][j] = in.nextInt();

}

}

System.out.println("Matrix 1:");

for (i = 0; i < r1; i++)

{

for (j = 0; j < c1; j++)

{

System.out.print(mat1[i][j]+"\t");

}

System.out.print("\n");

}

System.out.println("\nEnter the elements of the second matrix");

for ( i= 0 ; i < r2 ; i++ )

{

for ( j= 0 ; j < c2 ;j++ )

mat2[i][j] = in.nextInt();

}

System.out.println("Matrix 2:");

for (i = 0; i < r2; i++)

{

for (j = 0; j < c2; j++)

{

System.out.print(mat2[i][j]+"\t");

}

System.out.print("\n");

}

System.out.println("\nOutput Matrix:-");

for ( i= 0 ; i < r1 ; i++ )

for ( j= 0 ; j <c2;j++)

{

sum=0;

for ( k= 0 ; k <r2;k++ )

{

sum +=mat1[i][k]\*mat2[k][j] ;

}

res[i][j]=sum;

}

for ( i= 0 ; i < r1; i++ )

{

for ( j=0 ; j < c2;j++ )

{

System.out.print(res[i][j]+"\t");

}

System.out.println("\n");

}

}

else

System.out.print("Multipication cannot be done ");

}

}

**Output**

**Case1:**

Enter the number of rows of the first matrix

3

Enter the number columns of the first matrix

4

Enter the number of rows of the second matrix

2

Enter the number of columns of the second matrix

5

Multipication cannot be done

**Case2:**

Enter the number of rows of the first matrix

2

Enter the number columns of the first matrix

2

Enter the number of rows of the second matrix

2

Enter the number of columns of the second matrix

2

Enter the elements of the first matrix

3

4

2

3

Matrix 1:

3 4

2 3

Enter the elements of the second matrix

5

3

6

4

Matrix 2:

5 3

6 4

Output Matrix:-

39 25

28 18

**Result**

The following code has been executed successfully

**Exp. No:4 Date: 8/12/2021**

**Aim**

Write a Java program that creates a class named 'Employee' having the following members: Name, Age, Phone number, Address, Salary. It also has a method named 'print-Salary( )' which prints the salary of the Employee. Two classes 'Officer' and 'Manager' inherits the 'Employee' class. The 'Officer' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign

a name, age, phone number, address and salary to an officer and a manager by making an object of both of these classes and print the same.

**Algorithm**

Step 1: Start

Step 2: Create a class Employee with all the data members and methods required

Step 3: Create two sub classes (officer and Manager) which inherit from the parent class Employee

Step 4: Make an object for two sub classes and access the method in superclass

Step 5: Stop

**Source Code**

import java.util.\*;

import java.io.\*;

class Employee

{

String Name;

int age;

long phno;

String add;

int sal;

public void printsal()

{

System.out.println("Name: ”+Name);

System.out.println("Age- “+age);

System.out.println("Contact Number: “+phno);

System.out.println("Address: “+add);

System.out.println("Salary is ” + sal);

}

}

class Officer extends Employee

{

String spec;

public void prints()

{

System.out.println(“Specialized in “+spec);

}

}

class Manager extends Employee

{

String dept;

public void printd()

{

System.out.println(“Department- “+dept);

}

}

public class Sal

{

public static void main(String[] args)

{

Officer a = new Officer();

a.Name = "Hari";

a.age = 24;

a.phno =89343453;

a.add= "G1 chitranagar";

a.sal = 65000;

a.spec = "Data Analytics";

Manager b = new Manager();

b.Name = "praveen";

b.age = 28;

b.phno =89302585;

b.add= "G2 Mangalassery";

b.sal = 95000;

b.dept = "Fire and Safety";

System.out.println(“Officer:”);

a.printsal();

a.prints();

System.out.println(“\n”);

System.out.println(“Manager:”);

b.printsal();

b.printd();

}

}

**Output**

Officer:

Name: Hari

Age-44

Contact Number:89343453

Address:G1 chitranagar

Salary is 65000

Specialized in Data Analytics

Manager:

Name: Praveen

Age-35

Contact Number:89302585

Address:G2 Mangalassery

Salary is 95000

Department-Fire and Safety

**Result**

The following code have been executed successfully

**Exp. No:5 Date: 15/12/2021**

**Aim**

Write a java program to create an abstract class named Shape that contains an empty method

named numberOfSides( ). Provide three classes named Rectangle, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides( ) that shows the number of sides in the given geometrical structures.

**Algorithm**

Step 1: Start

Step 2: Create a super class Shape with an empty method

Step 3: Create three sub classes (rectangle, triangle and hexagon) inheriting from the super class

Step 4: Each sub class contains the same method defined with a print statement telling its number of sides

Step 5: Create an object for each sub class

Step 6: Call the method for each sub class to show the number of sides of shapes

Step 7: Stop

**Source Code**

import java.util.\*;

import java.io.\*;

class Shape

{

void numberOfSides()

{}

}

class Rectangle extends Shape

{

void numberOfSides()

{

System.out.println("Number of sides in Rectangle is 4");

}

}

class Triangle extends Shape

{

void numberOfSides()

{

System.out.println("Number of sides in Triangle is 3");

}

}

class Hexagon extends Shape

{

void numberOfSides()

{

System.out.println("Number of sides in Hexagon is 6");

}

}

public class shapeDemo

{

public static void main(String[] args)

{

Shape s=new Shape();

s = new Rectangle();

s.numberOfSides();

s=new Triangle();

s.numberOfSides();

s=new Hexagon();

s.numberOfSides();

}

}

**Output**

Number of sides in Rectangle is 4

Number of sides in Triangle is 3

Number of sides in Hexagon is 6

**Result**

The following code have been executed successfully.

**Exp. No:6 Date: 5/1/2022**

**Aim**

Write a java program to demonstrare the use of garbage collector.

**Algorithm**

Step 1: Start

Step 2: Create class named GC

Step 3: Create finalize method and print a statement

Step 4: Create main function

Step 5: Create object of GC

Step 6: Assign null to the object to unreference it

Step 7: Call system.gc

Step 8: Stop

**Source Code**

public class GC

{

protected void finalize()

{

System.out.println(“Checking GC”);

}

public static void main(String args[ ])

{

GC t=new GC();

t=null;

System.gc();

}

}

**Output**

Checking GC

**Result**

The following code have been executed successfully.

**Exp. No:7 Date:19/1/2022**

**Aim**

Write a file handling program in java with reader/writer.

**Algorithm**

Step 1: Start

Step 2: Create class FileReaderWriter

Step 3: Create a FileWriter object fw and the file to which it is written as “demo.txt”

Step 4: Write content into “demo.txt” file using write method and close it

Step 5: Create a FileReader instance fr of file “demo.txt”

Step 6: Print the contents of the file

Step 7: Try step 2 to step 7

Step 8: If any exception occurs, catch and print it

Step 9: Stop

**Source Code**

import java.io.\*;

public class fileReaderWriter {

public static void main(String args[]){

try

{

FileWriter fw=new FileWriter("D:\\demo.txt");

fw.write("Java programming");

fw.close();

FileReader fr=new FileReader("D:\\demo.txt");

int i;

while ((i = fr.read()) != -1)

{

System.out.print((char)i);

}

fr.close();

}

catch(Exception e)

{

System.out.println(e);

System.out.println("error");

}

System.out.println("\nSuccess...");

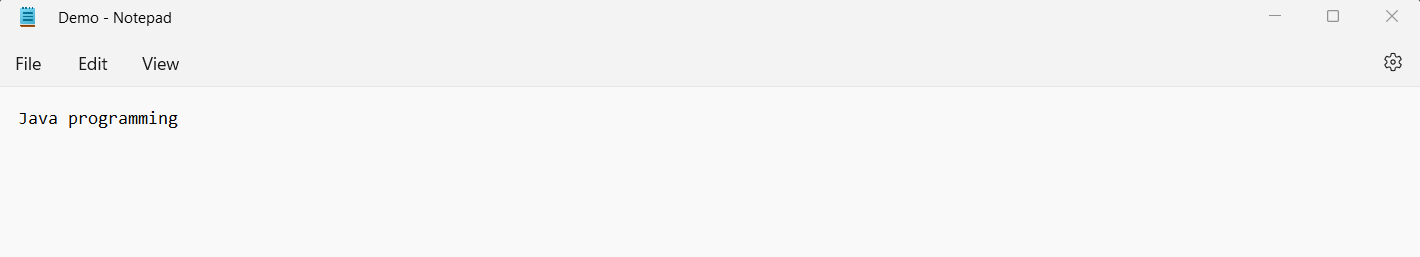
}

}

**Output**

Java programming

Success...



**Result**

The following code have been executed successfully.

**Exp. No: 8 Date: 19/1/2022**

**Aim**

Write a Java program that read from a file and write to file by handling all file related exceptions.

**Algorithm**

Step 1: Start

Step 2: Create a class Copy with functions void reading() and void copying(), both which throws IOException.

Step 3: Create a class trial and declare main method, which throws IOException.

Declare object test of class Copy. Call copying() using test to copy one file to another.

Display "Succesfully copied content from orginal to new " Call reading() using test to read the

copied file

Step 4: Stop

**Algorithm for reading()**

Step 1: Start

Step 2: Declare object fr of class FileReader and initialise to null.

Step 3: Inside try block:

Open trycopy.txt using fr.

Store each character of content in trycopy.txt in i using read(), and display it

using while loop.

Inside catch block with Exception e:

Display “The following exception has occured:-" and display e.

Inside finally block:

IF (fr!=null)

fr.close()

Step 4: Stop

**Algorithm for copying()**

Step 1: Start

Step 2: Declare object fw of class FileWriter and initialise to null.

Step 3: Declare object fr of class FileReader and initialise to null.

Step 4: Inside try block:

Open try.txt using fr.

Open trycopy.txt using fw.

Store each character of content in try.txt in i using read() by using while loop,

and write i in trycopy.txt using write().

fw.close()

Inside catch block with Exception e:

Display “The following exception has occured:-" and display e.

Inside finally block:

IF (fr!=null)

fr.close()

IF (fw!=null)

fw.close()

Step 5: Stop

**Source Code**

import java.io.\*;

import java.util.\*;

import java.lang.Exception;

class Copy

{

public void reading() throws IOException

{

FileReader fr=null;

try

{

fr=new FileReader("trycopy.txt");

int i;

System.out.println("The content of file is:");

while ((i=fr.read()) != -1){

System.out.print((char) i);

}

catch (Exception e)

{

System.out.println("The following exception has occured:-");

System.out.println(e);

}

finally

{

if (fr!=null)

{

fr.close();

}

}

}

public void copying() throws IOException

{

FileWriter fw=null;

FileReader fr=null;

try

{

fr=new FileReader("try.txt");

fw=new FileWriter("trycopy.txt");

int i;

while ((i=fr.read()) != -1)

fw.write(i);

}

catch (Exception e)

{

System.out.println("The following exception has occured:-");

System.out.println(e);

}

finally

{

if (fr!=null) {

fr.close();

}

if (fw!=null) {

fw.close();

}

}

}

}

class trial

{

public static void main(String[] args) throws IOException

{

System.out.println("COPYING ONE FILE TO ANOTHER BY HANDLING FILE

EXCEPTIONS");

System.out.println("-------------------------------------------------------");

System.out.println();

Copy test=new Copy();

System.out.println("Let us copy a file using FileReader and FileWriter");

test.copying();

System.out.println("Succesfully copied content from orginal to new");

System.out.println();

System.out.println("Let us now read the copied file using FileReader");

test.reading();

}

}

**Output**

COPYING ONE FILE TO ANOTHER BY HANDLING FILE EXCEPTIONS

-------------------------------------------------------

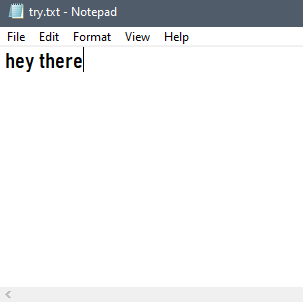
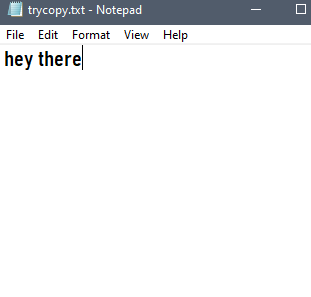
Let us copy a file using FileReader and FileWriter

Succesfully copied content from orginal to new

Let us now read the copied file using FileReader

The content of file is:

hey there

**Result**

The following code have been executed successfully.

**Exp. No: 9**  **Date: 19/1/2022**

**Aim**

Write a Java program that shows the usage of try, catch, throws and finally for Employee

Class.

**Algorithm**

Step 1: Start.

Step 2: Create Employee class and declare variables emp\_id, name, age, address, salary.

Step 3: Create a method that throws an ArithmeticException.

Step 4: Create class Emp.

Step 5: Create object for Employee class and Scanner class.

Step 6: Ask user input for emp\_id, name, address, age, salary.

Step 7: Using try block calculate salary to bonus ratio.

Step 8: Use catch block to catch exception.

Step 9: Use finally to display message.

Step 10: Throw an exception if age>=60.

Step 11: Stop.

**Source Code**

import java.io.\*;

import java.util.\*;

class Employee{

String emp\_id, name, address;

int age;

double salary;

void bon(int b) throws ArithmeticException{

System.out.println(salary/b);

}

}

class Emp{

public static void main(String args[]){

Employee employee = new Employee();

Scanner sc = new Scanner(System.in);

System.out.print("Emp.ID: ");

employee.emp\_id = sc.nextLine();

System.out.print("Name: ");

employee.name = sc.nextLine();

System.out.print("Address: ");

employee.address = sc.nextLine();

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

employee.age = sc.nextInt();

System.out.print("Salary: ");

employee.salary = sc.nextDouble();

System.out.print("Bonus: ");

int b = sc.nextInt();

try{

System.out.println("Salary to bonus ratio");

employee.bon(b);

}

catch(ArithmeticException e){

System.out.println("Bonus should not be 0");

}

finally{

System.out.println("Finally is executed");

if(employee.age >= 60){

throw new

ArithmeticException("retired");

}

else{

System.out.println("Never Quit");

}

System.out.println("Have a nice day");

}

sc.close();

}

}

**Output**

Emp.ID: 007

Name: Joe

Address: Hazel villa, Street no 5, Banglore

Age: 27

Salary: 80000

Bonus: 2000

Salary to bonus ratio

20.0

Finally, is executed

Never Quit

Have a nice day

**Result**

The following code has been executed successfully.

**Exp. No: 10 Date: 2/2/2022**

**Aim**

Write a Java program that implements a multi-threaded program which has three threads. First thread generates a random integer every 1 second. If the value is even, second thread computes the square

of the number and prints. If the value is odd the third thread will print the value of cube of the number.

**Algorithm**

Step1: Start

Step2: Create classes for squaring and cubing numbers that extend thread and write

constructors for the same and also write the run method for printing the respective data.

Step3: Create a class extending Thread that selects a random number and has objects from the

first two classes inside the run method to execute the necessary operation

Step4: inside main class create an object for the number class and start the run method

Step5: Stop

**Source Code**

import java.util.Random;

class Square extends Thread{

int x;

Square(int x){

this.x = x;

}

public void run(){

System.out.println("The square of " + x +" is " + x\*x);

}

}

class Cube extends Thread{

int x;

Cube(int x){

this.x = x;

}

public void run(){

System.out.println("The cube of " + x + " is " + x\*x\*x);

}

}

class Number extends Thread{

public void run(){

Random r = new Random();

for(int i=0;i<5;i++){

int x = r.nextInt(100);

System.out.println("The random number is " + x);

if(x%2 == 0){

Square s = new Square(x);

s.start();

}else{

Cube c = new Cube(x);

c.start();

}

try{

Thread.sleep(1000);

}

catch(InterruptedException e){

System.out.println("Thread interrupted");

}

}

}

}

class Thmain{

public static void main(String[] args){

Number n = new Number();

n.start();

}

}

**Output**

The random number is 78

The square of 78 is 6084

The random number is 81

The cube of 81 is 531441

The random number is 64

The square of 64 is 4096

The random number is 19

The cube of 19 is 6859

The random number is 44

The square of 44 is 1936

**Result**

The Program was executed successfully.

**Exp. No: 11 Date: 2/2/2022**

**Aim**

Write a Java program that shows thread synchronization

**Algorithm**

Step1: Start

Step2: create a class with synchronized method to find the tables of the number till the 10s

Step3: create a first thread class extending thread class which creates an object for the tables

Class and prints the tables of 5 with a 1 second wait

Step4: create a second thread class which does the same but with 7

Step5: Execute both threads in main class

Step6: Stop

**Source code**

class Tables{

synchronized void printTable(int n){

for(int i=1;i<11;i++){

System.out.println(n\*i);

try{

Thread.sleep(1000);

}

catch(InterruptedException e){

System.out.println("Thread interrupted.");

}

}

}

}

class Thread1 extends Thread{

Tables t;

Thread1(Tables t){

this.t=t;

}

public void run(){

t.printTable(5);

}

}

class Thread2 extends Thread{

Tables t;

Thread2(Tables t){

this.t=t;

}

public void run(){

t.printTable(7);

}

}

class Thsync {

static void main(String[] args){

Tables t=new Tables();

Thread1 t1=new Thread1(t);

Thread2 t2=new Thread2(t);

t1.start();

t2.start();

}

}

**Output**

5

10

15

20

25

30

35

40

45

50

7

14

21

28

35

42

49

56

63

70

**Result**

Program was executed successfully

**Exp.No:12 Date:09/11/2020**

**Aim**

Write a Java program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util).

**Algorithm**

Step 1:Start

Step 2:Read n, sum=0

Step 3:Read the set of integers

Step 4:Initialize object(st) for string tokenizer

Step 5: while (st.hasMoreTokens())

Read n sum=sum+n

Print n

Step 6:Print sum

Step 7:Stop

**Source Code**

import java.util.\*;

public class StringTokenizerExample

{

public static void main(String args[])

{

int n;

int sum = 0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter integers in a sequence:");

String s = sc.nextLine();

StringTokenizer st = new StringTokenizer(s, " ");

while (st.hasMoreTokens()) {

String temp = st.nextToken();

n = Integer.parseInt(temp);

System.out.println(n);

sum = sum + n;

}

System.out.println("sum of the integers is: " + sum);

sc.close();

}

}

**Output**

Enter the integers in sequence:

20 67 13

20

67

13

Sum of the integers is: 100

**Result**

The following program has executed successfully.

**Exp.No:13 Date 9/2/2022**

**Aim**

Write a java program for the following:

a)Create a doubly linked list of elements

b)Delete a given element from the above list

c)Display content of list after deletion

**Algorithm**

Step1:start

Step2:create class dll

Step3: :create linked list object of integer type as dll

Step4: use add(element e) method to add elements into dll

Step5:iterate through each element of dll and print it

Step6:take user input of the element to deleted through object of scanner class

Step7:use remove(int index) method to remove user input item from dll

Step8:print dll

Step9:stop

**Source code**

Import java.util.\*;

Public class dll

{

Public static void main(String args[])

{

Linkedlist<Integer>dll=new Linkedlist<Integer>();

dll.add(1);

dll.add(2);

dll.add(3);

Iterator<Integer>i=dll.iterator();

System.out.println(“Elements of dll”);

While(i.haNext())

{

System.out.prinltn(i.next());

}

System.out.println(“Enter the position of the item to be deleted’);

Scanner sc=new Scanner(System.in);

Int a=sc.nextInt();

dll.remove(a);

System.out.println(“Elements of dll after deletion:”);

System.out.println(dll);

}

}

**Output**

Elements of dll

1

2

3

Enter the position of item to be deleted:1

Elements of dll after deletion:[1,3]

**Result**

The following program has executed successfully.

**Exp.No:14 Date:16/02/2022**

**Aim**

Write a Java program that works as a simple calculator. Arrange Buttons for digits and the +,-, \*, % operations properly. Add a text field to display the result. Handle any possible exceptions like divide by zero. Use Java Swing.

**Algorithm**

Step 1: Start

Step 2: Create a class Calculator extends Jframe implements ActionListener

Step 3: Create three labels for entering two numbers and displaying result.

Create six buttons for 4 operations and Reset , Clear and 3 textfields to display.

Step 4: In the constructor of class Calculator

Give name for the Frame.

Initialize all the labels, textfields and buttons.

Add the components to the Frame.

Set the size, layout and visibility of the Frame.

Register the button Components to the Source.

Step 5: In the actionPerformed method,

Declare two string to get the input from the user using two textfield object.

In the Try block ,

Check the button clicked and perform the appropriate operation.

Display the result in the textfield.

In the catch block

Catch the exception that may have occurred and display it.

Step 6: In the main() method

Call the constructor of Calculator class.

Step 7: Stop

**Source code**

import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.lang.Exception;

public class Calculator1 extends JFrame implements ActionListener

{

JLabel l1,l2,l3;

JTextField t1,t2,t3;

JButton b1,b2,b3,b4,b5,b6;

public Calculator1()

{

l1=new JLabel("Num 1");

l2=new JLabel("Num 2");

l3=new JLabel("Result");

t1=new JTextField(20);

t2=new JTextField(20);

t3=new JTextField(20);

b1=new JButton("+");

b2=new JButton("-");

b3=new JButton("\*");

b4=new JButton("/");

b5=new JButton("Reset");

b6=new JButton("Close");

add(l1);

add(t1);

add(l2);

add(t2);

add(l3);

add(t3);

add(b1);

add(b2);

add(b3);

add(b4);

add(b5);

add(b6);

setSize(300,300);

setLayout(new FlowLayout());

setVisible(true);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

b5.addActionListener(this);

b6.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

String s1=t1.getText();

String s2=t2.getText();

double a=0,b=0,c=0;

try

{

if(ae.getSource()==b1)

{

a=Double.parseDouble(s1);

b=Double.parseDouble(s2);

c=a+b;

t3.setText(String.valueOf(c));

}

else if(ae.getSource()==b2)

{

a=Double.parseDouble(s1);

b=Double.parseDouble(s2);

c=a-b;

t3.setText(String.valueOf(c));

}

else if(ae.getSource()==b3)

{

a=Double.parseDouble(s1);

b=Double.parseDouble(s2);

c=a\*b;

t3.setText(String.valueOf(c));

}

else if(ae.getSource()==b4)

{

a=Double.parseDouble(s1);

b=Double.parseDouble(s2);

c=a/b;

t3.setText(String.valueOf(c));

}

else if(ae.getSource()==b5)

{

t1.setText("0");

t2.setText("0");

t3.setText("0");

}

else

{

System.exit(0);

}

}

catch(Exception e)

{

System.out.println("The following exception occurred during execution");

System.out.println(e);

}

}

public static void main(String args[])

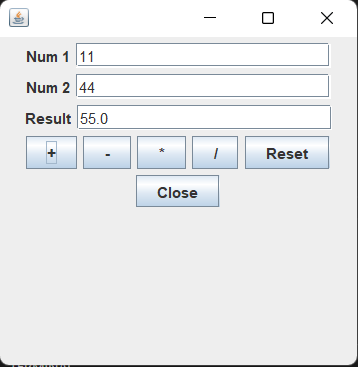
{

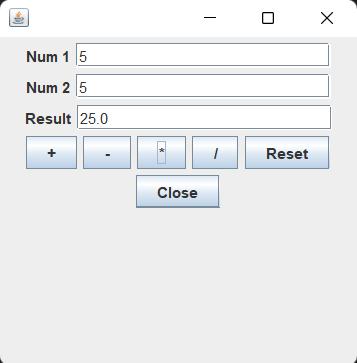
new Calculator1();

}

}

**Output**





**Result**

The following program has executed successfully.

**Exp.No:15 Date:16/02/2022**

**Aim**

Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts.

**Algorithm**

Step 1: Start

Step 2: Create 3 circles, colored red , yellow and green and create corresponding buttons

Step 3: Dim all the colors , when the button is clicked corresponding color is shown at full brightness

Step 4: Stop

**Source code**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class trafficlight extends JFrame implements ActionListener

{

JRadioButton r,g,y;

Container c;

ButtonGroup bg;

public trafficlight()

{

r=new JRadioButton("RED");

g=new JRadioButton("YELLOW");

y=new JRadioButton("GREEN");

c= getContentPane();

bg= new ButtonGroup();

add(r);add(y);add(g);

bg.add(r);bg.add(y);bg.add(g);

c.add(r);c.add(g);c.add(y);

setLayout(new FlowLayout());

setVisible(true);

setSize(200,200);

r.addActionListener(this);

g.addActionListener(this);

y.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

String msg=ae.getActionCommand();

if(msg.equals("RED"))

{

c.setBackground(Color.RED);

}

if(msg.equals("YELLOW"))

{

c.setBackground(Color.YELLOW);

}

if(msg.equals("GREEN"))

{

c.setBackground(Color.GREEN);

}

}

public static void main(String[] args)

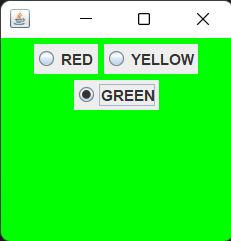
{

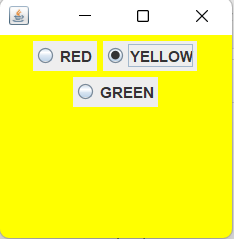
trafficlight t=new trafficlight();

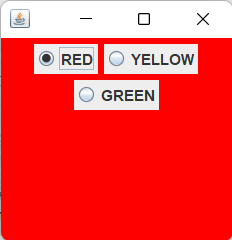
}

}

**Output**







**Result**

The following program has executed successfully.

**Exp.No:16 Date:16/02/2022**

**Aim**

Create a java program to connect to jdbc and display all the tables

**Algorithm**

Step 1: Start

Step 2: Create a connection to sql database

Step 3: Insert data after creating a table

Step 4: Display the contents of the table

Step 5: Stop

**Source Code**

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.Statement;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.border.EmptyBorder;

public class UserRegistration extends JFrame {

private static final long serialVersionUID = 1L;

private JPanel contentPane;

private JTextField firstname;

private JTextField state;

private JTextField mob;

private JButton btnNewButton;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

UserRegistration frame = new UserRegistration();

frame.setVisible(true);

}

catch (Exception e) {

e.printStackTrace();

}

}

}

}

public UserRegistration() {

setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\User\\Desktop\\STDM.jpg"));

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(450, 190, 1014, 597);

setResizable(false);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

contentPane.setBackground(Color.decode("#99e2b4"));

setContentPane(contentPane);

contentPane.setLayout(null);

JLabel lblNewUserRegister = new JLabel("Covid Form");

lblNewUserRegister.setFont(new Font("Times New Roman", Font.PLAIN, 42));

lblNewUserRegister.setBounds(362, 52, 325, 50);

contentPane.add(lblNewUserRegister);

JLabel lblName = new JLabel("Name");

lblName.setFont(new Font("Tahoma", Font.PLAIN, 20));

lblName.setBounds(58, 152, 150, 43);

contentPane.add(lblName);

firstname = new JTextField();

firstname.setFont(new Font("Tahoma", Font.PLAIN, 32));

firstname.setBounds(264, 151, 248, 50);

contentPane.add(firstname);

firstname.setColumns(10);

state = new JTextField();

state.setFont(new Font("Tahoma", Font.PLAIN, 32));

state.setBounds(264, 243, 248, 50);

contentPane.add(state);

state.setColumns(10);

JLabel lblstate = new JLabel("Are you covid positive");

lblstate.setFont(new Font("Tahoma", Font.PLAIN, 20));

lblstate.setBounds(58, 243, 200, 49);

contentPane.add(lblstate);

JLabel lblMobileNumber = new JLabel("Mobile number");

lblMobileNumber.setFont(new Font("Tahoma", Font.PLAIN, 20));

lblMobileNumber.setBounds(58, 329, 139, 26);

contentPane.add(lblMobileNumber);

mob = new JTextField();

mob.setFont(new Font("Tahoma", Font.PLAIN, 32));

mob.setBounds(264, 320, 248, 50);

contentPane.add(mob);

mob.setColumns(10);

btnNewButton = new JButton("Submit");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String firstName = firstname.getText();

String statement = state.getText();

String mobileNumber = mob.getText();

int len = mobileNumber.length();

String msg = "" + firstName;

msg += " \n";

if (len != 10) {

JOptionPane.showMessageDialog(btnNewButton, "Enter a valid mobile number");

}

try {

Connection connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/cowin\_demo", "root", "root");

String query = "INSERT INTO form values('" + firstName + "','" + statement + "','" + mobileNumber + "')";

Statement sta = connection.createStatement();

int x = sta.executeUpdate(query);

if (x == 0) {

JOptionPane.showMessageDialog(btnNewButton, "This is alredy exist");

}

else {

JOptionPane.showMessageDialog(btnNewButton,msg + "You have sucessfully

submited");

}

connection.close();

}

catch (Exception exception) {

exception.printStackTrace();

}

}

});

btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 22));

btnNewButton.setBounds(399, 447, 259, 74);

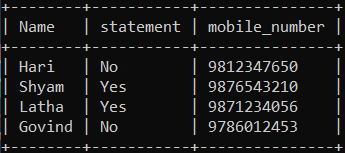
contentPane.add(btnNewButton);

}

}

**Output**





**Result**

The following program has executed successfully.