**Implement a Java program that creates two threads. One thread should print even numbers, and the other should print odd numbers from 1 to 10.**

package thread;

public class Threadings {

class A extends Thread {

public void run() {

for (int i = 0; i < 10; i++) {

if (i % 2 == 0) {

System.out.println("Even : " + i);

}

}

}

}

class B extends Thread {

public void run() {

for (int i = 0; i < 10; i++) {

if (i % 2 != 0) {

System.out.println("Odd: " + i);

}

}

}

}

public static void main(String[] args) {

Threadings threadings = new Threadings(); // create an instance of the outer class

A a = threadings.new A(); // create an instance of class A

B b = threadings.new B(); // create an instance of class B

try {

a.sleep(100);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

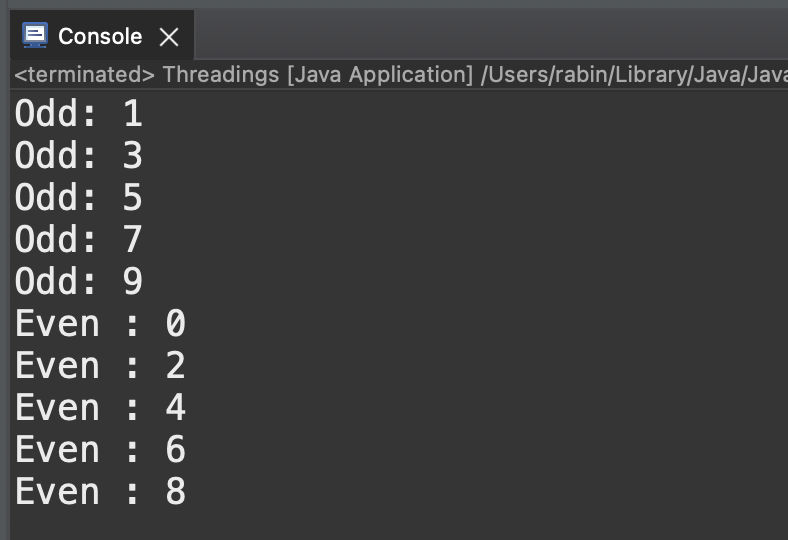
a.start(); // start the thread for class A

b.start(); // start the thread for class B

}

}

**OUTPUT:**

****

**Create a Java program with two threads sharing a common resource (e.g., a counter). Implement synchronization to ensure that the threads alternate incrementing the counter.**

package thread;

public class SharedResourceExample {

private static final int MAX\_COUNT = 5;

private static int counter = 0;

public static void main(String[] args) {

// Create two threads

Thread thread1 = new Thread(new IncrementTask());

Thread thread2 = new Thread(new IncrementTask());

// Start the threads

thread1.start();

thread2.start();

try {

// Wait for both threads to finish

thread1.join();

thread2.join();

} catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println("Final counter value: " + counter);

}

static class IncrementTask implements Runnable {

@Override

public void run() {

for (int i = 0; i < MAX\_COUNT; i++) {

synchronized (SharedResourceExample.class) {

// Increment the counter

counter++;

System.out.println(Thread.currentThread().getName() + ": Counter = " + counter);

}

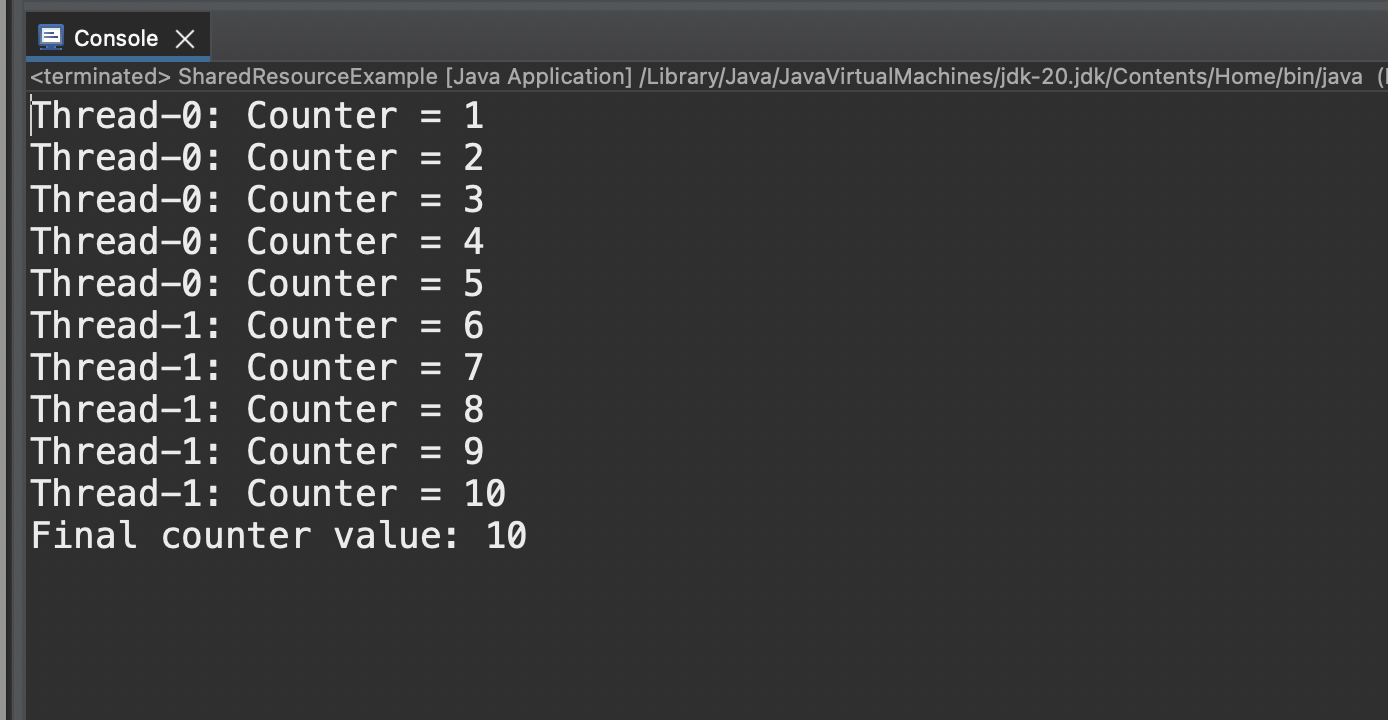
}

}

}

}

**OUTPUT:**

****

**Develop a Java program that creates three threads with different priorities.**

package thread;

public class ThreadPriorityExample {

public static void main(String[] args) {

PriorityThread thread1 = new PriorityThread("Thread 1");

PriorityThread thread2 = new PriorityThread("Thread 2");

PriorityThread thread3 = new PriorityThread("Thread 3");

// Set thread priorities

thread1.setPriority(Thread.MIN\_PRIORITY); // Lowest priority (1)

thread2.setPriority(Thread.NORM\_PRIORITY); // Default priority (5)

thread3.setPriority(Thread.MAX\_PRIORITY); // Highest priority (10)

// Start the threads

thread1.start();

thread2.start();

thread3.start();

// Wait for all threads to finish

try {

thread1.join();

thread2.join();

thread3.join();

} catch (InterruptedException e) {

e.printStackTrace();

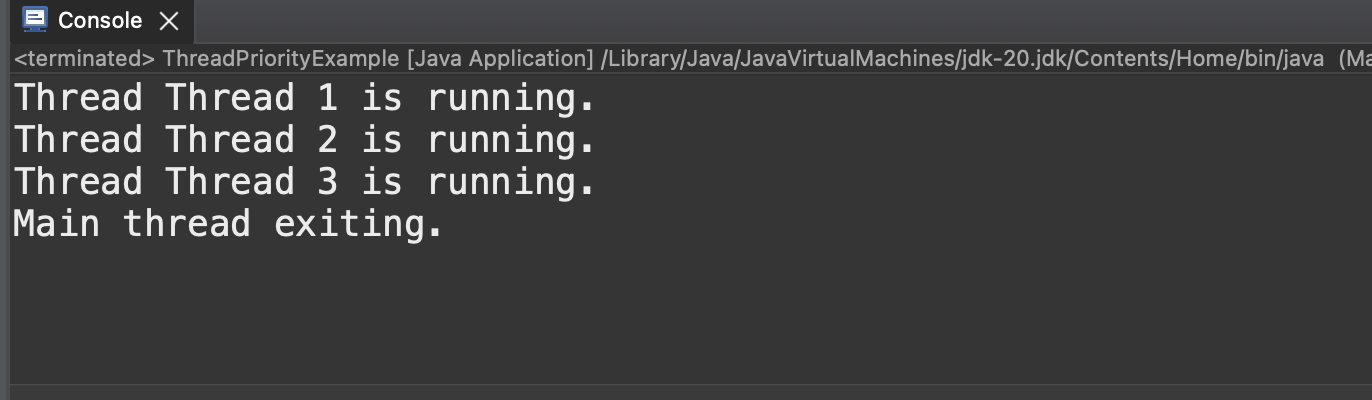
}

System.out.println("Main thread exiting.");

}

}

OUTPUT:



**Create a Java program that reads data from a text file and displays it on the console. Ensure**

**Proper exception handling.**

package File\_Handling;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

public class BufferReaders {

public static void main(String[] args) {

char[] array = new char[100];

try {

// creates a file reader

FileReader readFile = new FileReader("../Classroom/src/bufferwriter.txt");

// Creates a buffer reader

BufferedReader buffers = new BufferedReader(readFile);

// Reads characters

try {

System.out.println("Data in the Stream: ");

buffers.read(array);

} catch (IOException e) {

e.printStackTrace();

}

System.out.println(array);

} catch (FileNotFoundException e) {

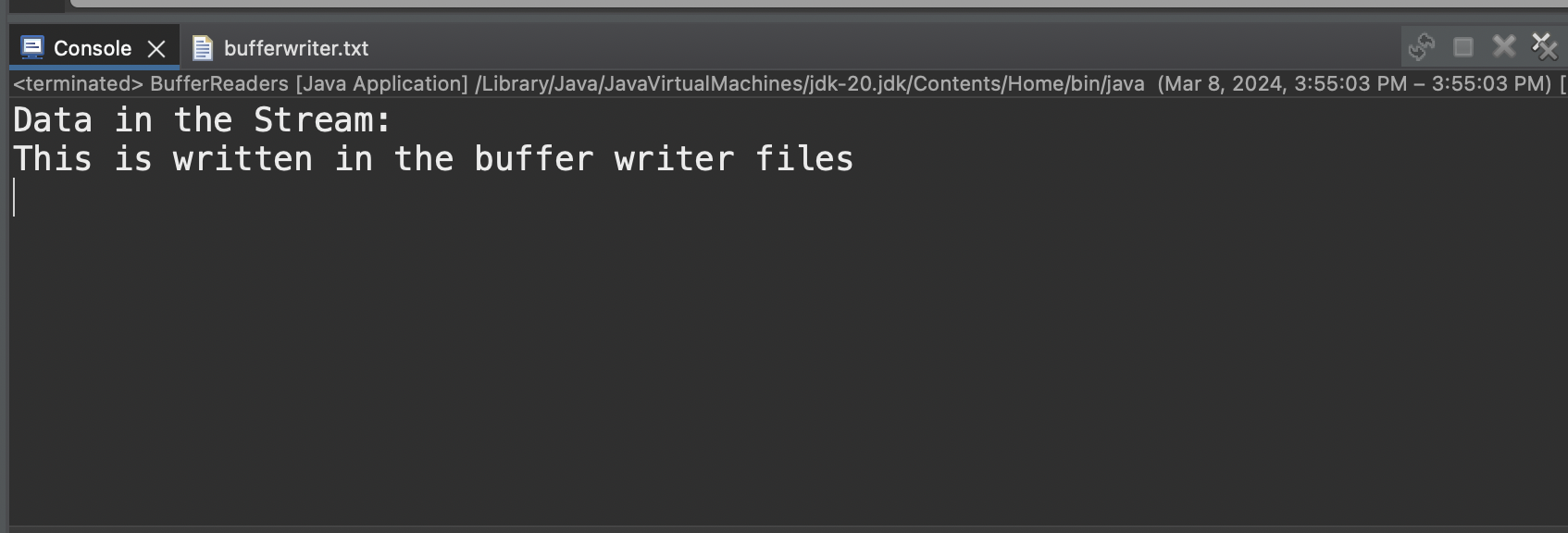
e.printStackTrace();

}

}

}

**OUTPUT:**



**Write a Java program to copy the contents of one text file to another new file.**

package File\_Handling;

import java.io.BufferedReader;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.OutputStreamWriter;

public class Source\_to\_Destination {

public static void main(String[] args) {

try {

// Read data from the source file

BufferedReader reader = new BufferedReader(new FileReader("../Classroom/src/bufferwriters.txt"));

StringBuilder stringBuilder = new StringBuilder();

String line;

while ((line = reader.readLine()) != null) {

stringBuilder.append(line);

stringBuilder.append(System.*lineSeparator*());

}

reader.close();

String data = stringBuilder.toString();

// Creates a FileOutputStream

FileOutputStream file = new FileOutputStream("../Classroom/src/destination.txt");

// Creates an OutputStreamWriter

OutputStreamWriter output = new OutputStreamWriter(file);

// Writes string to the file

output.write(data);

System.***out***.println("\n File Written Successfully");

// Closes the writer

output.close(); // close outputStream

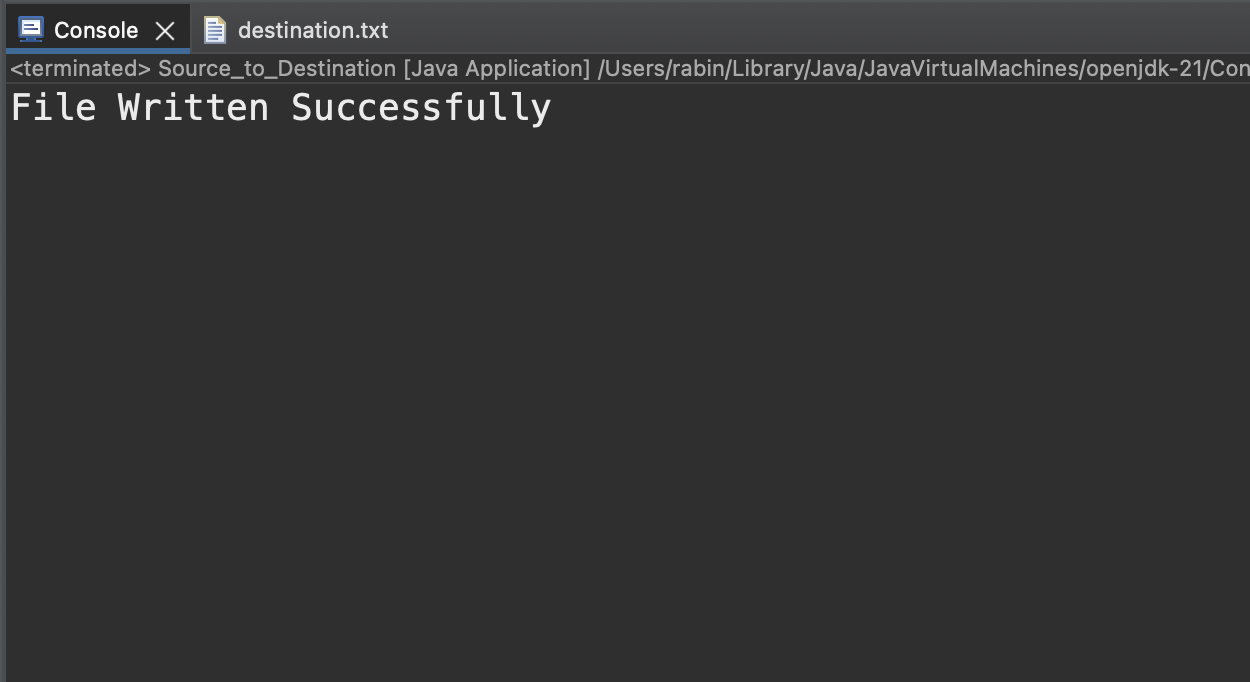
} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT**:  
 

**Design a simple login form using Java Swing components. Include JTextField for username, JPasswordField for password, and a JButton to submit the login. Display a message in a JLabel based on whether the login is successful or not.**

package Swing;

// imports

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import javax.swing.JButton;

import javax.swing.JPasswordField;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Id\_Password {

public static void main(String[] args) {

// frame objects

JFrame frame = new JFrame("Login Application");

JLabel name = new JLabel(" Name:");

JLabel password = new JLabel("Password: ");

JButton button = new JButton("Submit");

// create objects

JTextField textField = new JTextField();

JPasswordField passfield = new JPasswordField(); // create for passwordField

// position set Display

name.setBounds(10, 20, 150, 30); // name position

textField.setBounds(100, 20, 100, 30); // name field position

password.setBounds(10, 45, 150, 30); // Password position

passfield.setBounds(100, 45, 100, 30); // password field Position

button.setBounds(80, 100, 70, 50); // Button position

// Action Listener to Submit

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

JOptionPane.*showMessageDialog*(null, "Login successfully", "Success", JOptionPane.***INFORMATION\_MESSAGE***);

}

});

// add to frames

frame.add(name);

frame.add(textField);

frame.add(password);

frame.add(passfield);

frame.add(button);

frame.setLayout(null);

frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

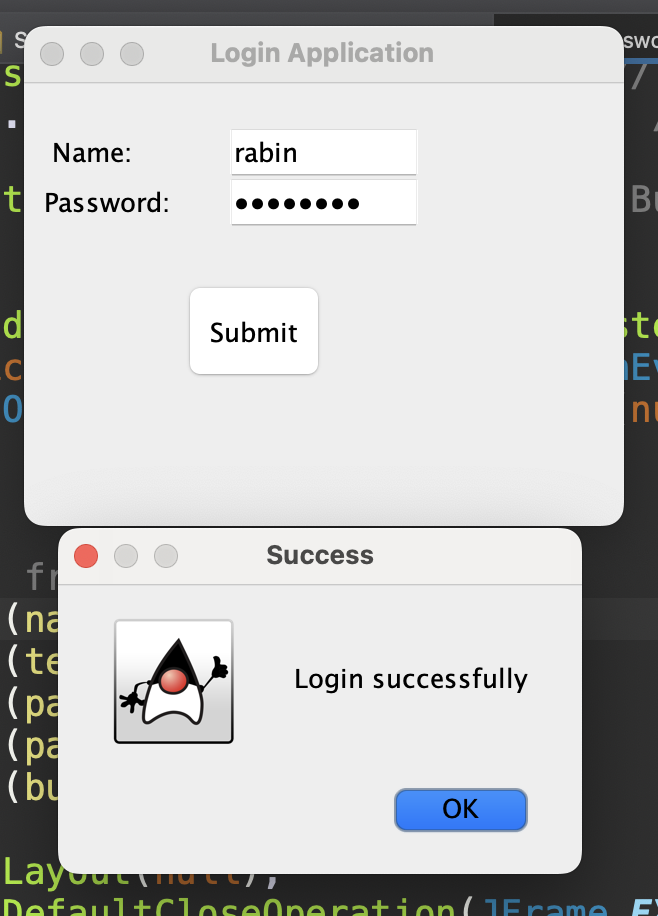
frame.setSize(300, 250);

frame.setVisible(true);

}

}

**OUTPUT**:



**Display the data from the Database**

package mysql;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class Select {

public static void main(String[] args) {

// connection

final String DRIVER = "com.mysql.cj.jdbc.Driver"; // Driver link provided

// Database connection

final String DBNAME = "JavaCollege"; // Database table name

final String HOST = "localhost"; // hosting site

final String DBUSER = "root"; // database name

final String DBPASS = "Neupane@11"; // password

final int PORT = 3306;

final String URL = "jdbc:mysql://" + HOST + ":" + PORT + "/" + DBNAME;

try {

// connection

Class.forName(DRIVER); // Loading Driver

Connection conn = DriverManager.getConnection(URL, DBUSER, DBPASS); // request for connection

// Insert Records

Statement state = conn.createStatement(); // object create for connection

String sql = "SELECT \* FROM Student";

ResultSet rs = state.executeQuery(sql); // Get all records from table

System.out.println("PID\t\t Name\t\tAddress");

while (rs.next()) {

System.out.println(rs.getInt("id") + "\t \t" + rs.getString("name") + "\t\t" + rs.getString("address"));

}

rs.close();

state.close();

conn.close();

System.out.print("Display all records successfully");

} catch (SQLException ex) {

System.out.println(ex);

}

catch (Exception ex) {

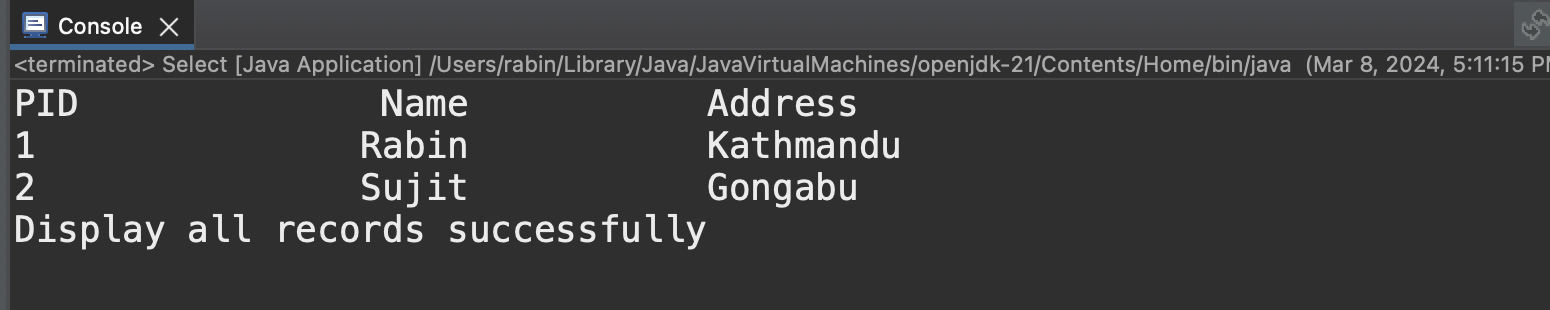
System.out.println("Error : " + ex.getMessage());

}

}

}

**OUTPUT:**

****