K L UNIVERSITY

FRESHMAN ENGINEERING DEPARTMENT

A Project Based Lab Report

On

Notes Making

SUBMITTED BY:

Name	Reg ID
A. Deepika	2200031890
K. Srikara Vinay	2200031897
K. Trisha Ramani	2200031928
S. V J S N S L P Khowshikk Sharma	2200031954

UNDER THE ESTEEMED GUIDANCE OF

V. Premalatha
Asst.Professor C.S.E

KL UNIVERSITY
Green fields, Vaddeswaram – 522502
Guntur Dt., AP, India.

DEPARTMENT OF BASIC ENGINEERING SCIENCES-1

CERTIFICATE

This is to certify that the project-based laboratory report entitled "Notes Making" submitted by Mr./Ms**. A. DEEPIKA, K. SRIKARA VINAY, K. TRISHA RAMANI, S. V J S N S L P KHOWSHIKK SHARMA** bearing Regd.ID.no **2200031890, 2200031897, 2200031928, 2200031954** to the **Department of Basic Engineering Sciences-1, KL University** in partial fulfilment of the requirements for the completion of a project-based Laboratory in **"Computational Thinking for Object Oriented Design"** course in I B Tech II Semester, is a bonafide record of the work carried out by BATCH-13 under my supervision during the academic year 2022 – 2023.

PROJECT SUPERVISOR

V. Premalatha

HEAD OF THE DEPARTMENT

Dr. Haritha

ACKNOWLEDGEMENTS

It is great pleasure to express my gratitude to our honourable President **Sri. Koneru Satyanarayana**, for giving me the opportunity and platform with facilities in accomplishing the project-based laboratory report.

I express my sincere gratitude to our Director **Dr. A. Jagdeesh** for his administration of our academic growth.

I express sincere gratitude to our Coordinator **Dr. M. Siva Ganga Prasad** and HOD-BES-1 **Dr. D. Haritha** for her leadership and constant motivation provided in the successful completion of our academic semester. I record it as my privilege to deeply thank you for providing us with the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor **V. Premalatha** for her novel association of ideas, encouragement, appreciation, and intellectual zeal which motivated us to venture into this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to making this project report successful.

ABSTRACT

The goal of this project is to create a Graphical User Interface (GUI) based Java program that allows users to log in using their credentials, view previously saved notes, add new notes, and delete existing notes.

The program will provide a login page that accepts user ID and password and will display all notes saved by the user. The user will be able to delete old notes and create new ones as well. The program is intended to be user-friendly, secure, and easy to use.

The program will use encryption to secure user data and will store user passwords in a hashed format to prevent them from being easily compromised. The program will also provide users with the option to customize the interface by choosing different cut, copy, open, save, new and print.

The notes will be stored in a database that can be accessed by the program. The user will be able to delete old notes and create new ones as well. The program will allow users to create new notes by typing text in a text field or by uploading files in different formats.

INDEX

INTRODUCTION

- To execute the program the concepts that have been used are GUI (Frames, Panels, Panes, Files, Text fields, Labels etc.), String (Length of the String, String equals), Conversion of data types (String to Double), Arrays (To store the details of the transactions). The user interface will be designed to be intuitive and easy to navigate, with clear instructions for each action.
- First create the Login page Frame using complete GUI concepts, so create a Frame with the title at the center as “NOTES” with two text fields for Username, Password with the correct you enter into the frame of file. Users will have the option to delete old notes and create new ones, ensuring that their notes remain organized and up-to-date.
- To create the graphical user interface, the program will utilize Swing, a Java-based framework. Swing provides a range of components for creating user interfaces, including buttons, text fields, and labels
- In summary, the proposed GUI-based Java program will enable users to manage their notes efficiently. With the ability to delete old notes and create new ones, users will be able to keep their notes organized and up-to-date. The program will use Java, SQL, and Swing to create a user-friendly interface that will make note-taking a breeze.
- The program features a simple and intuitive user interface, with a text area where users can type and edit their notes. The notes are saved as text files, which can be opened, edited, and saved using the program's file menu options.

- The program's interface is built using Java's Swing framework, which provides a set of graphical user interface components and tools for building desktop applications. The program also uses the AWT (Abstract Window Toolkit) and event handling classes to create and manage GUI components, handle user events, and perform various program tasks.
- The program's main window is a JFrame object that contains a JTextArea component and a JScrollPane component, which enables scrolling if the notes exceed the text area's dimensions. The program also includes a JMenuBar object that contains several JMenu and JMenuItem objects for opening, saving, printing, cutting, copying, and pasting notes, as well as closing the program and providing information about it.
- The program's actionPerformed method handles all user actions and performs the necessary actions based on the user's selection. For example, if the user selects "Save" from the file menu, the program creates a JFileChooser object and prompts the user to select a file to save the notes. The program then writes the contents of the JTextArea to the selected file using a FileWriter and BufferedWriter object.
- Overall, the M3_NOTES program is a simple and useful tool for creating, editing, and saving notes in a GUI-based environment. Its intuitive interface and easy-to-use features make it an ideal solution for users who want a simple and efficient note-taking application.

AIM OF PROJECT

Implement a GUI based java program which should have a login page and should take in ID and password as login credentials. Then it should display all the notes that were previously saved by the user. It should give him the option to delete previous notes and also add new notes.

Advantages: -

- Provides a graphical user interface (GUI) for users to interact with the program.
- Implements a login page to authenticate users, ensuring only authorized users can access the notes.
- Allows users to view, delete and add notes.
- Uses ImageIO and ImageIcon to load and display images in the program.
- Uses JPasswordField to securely handle password input.
- It provides a simple and user-friendly interface to create and edit text files.
- It allows users to perform basic file operations such as opening, saving, and printing files.
- It implements the ActionListener interface to handle user events such as button clicks and menu selections.
- It uses the JScrollPane class to provide a scrollable view of the text area.
- It includes a JMenuBar and multiple JMenu items to organize menu options.
- It uses try-catch blocks to handle exceptions that may occur during file operations.
- It uses the JFileChooser class to provide a file selection dialog for opening and saving files.
- It provides the ability to cut, copy, and paste text within the text area.
- It implements a feature to set the font of the text area.

- It includes an "About Notes" menu item to display information about the application.

Disadvantages: -

- The program does not implement any encryption or hashing techniques to secure the password during transmission or storage.
- The program stores note locally, which could be lost if the device is damaged, lost or stolen. Cloud-based storage solutions could be implemented for better data safety.
- The program uses null layout (absolute positioning) which can cause layout problems on different screen sizes or resolutions.
- The program lacks error handling, which could lead to unexpected crashes or behaviour if input errors or exceptions occur.
- The program does not provide any means for users to recover lost or forgotten passwords.
- It may be limited in functionality for advanced users who require more complex text editing features.
- It may not handle very large files efficiently, as it loads the entire file into memory.
- It does not provide an undo/redo feature for text editing.
- It does not include a spell-check feature.
- It uses the default Metal Look and Feel for the GUI, which may not be aesthetically pleasing to some users.

SYSTEM REQUIREMENTS

- **SOFTWARE REQUIREMENTS:**

The major software requirements of the project are as follows:

Language : JDK 17.0.2 ECLIPSE

Operating System : Windows 11.

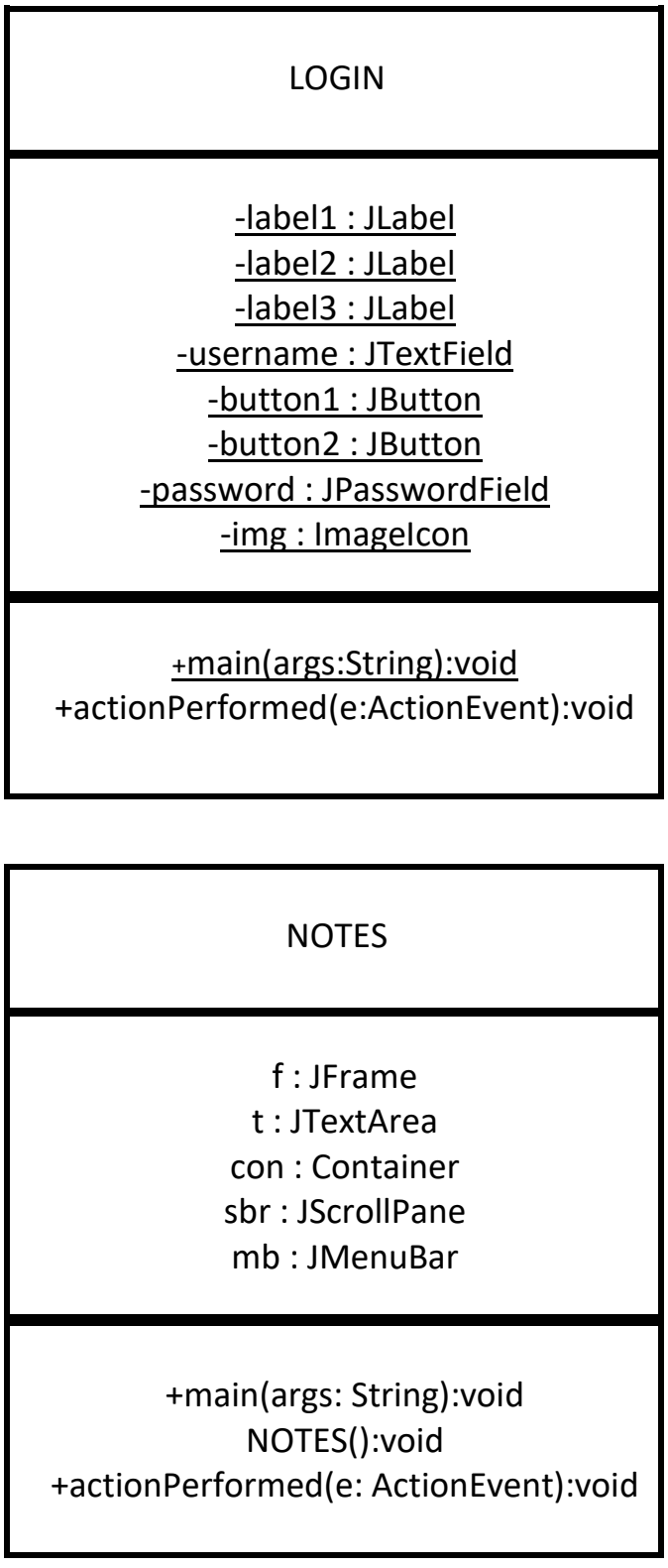
- **HARDWARE REQUIREMENTS:**

The hardware requirements that map to the software are as follows:

RAM : 8.00 GB

Processor : Intel(R) Core (TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz

CLASS DIAGRAM



ALGORITHM

Frame: 1

- Step-1: Create a class “LOGIN”.
- Step-2: import swing, awt, awt event which implements Action Listener and extends JFrame with throws IO Exception for main method.
- Step-3: Create a frame and add Panels In which the title as “Login page”, as it contains a picture of the logo of the Notes, and contains two JLabel that are Username, Password and two Jtextfield to enter the data and two JButtons b1, b2 to Login or to Reset
- Step-4: When clicked on b1 if credentials are right then the new file notes frame is opened after the message of “Login Successful. “is displayed, or else show message as “Wrong Username or Password”, When clicked on Reset, reset the entered credentials.

Frame: 2

- Step-1: Create a class as “NOTES”.
- Step-2: import swing, awt, awt, io, plaf. metal, text event classes which implements Action Listener and extends JFrame
- Step-3: Do the all operation in Constructer as NOTES () {..}.
- Step-4: Create a Frame named as “Notes” in Location of new points with size in dimension of user wanted and set font to the frame as in need with setting visibility as “true” and JTextArea for the text field to make notes, Container for all other Panes, JScrollPane created in vertical and horizontal in formatted of as needed, JMenu for file menu operations such as new, open, save and print.

- Step-5: In Menu create JMenuItem of File item contains of “New, Open, Save and Print”, Edit item contains of “Cut, Copy, Paste” and Help item contains of “About Notes and Close” which are in implements actionlistener.

- Step-6: Create a method of actionlistener as if else conditional statements used for JMenuItem’s with the JFileChooser, FileReader, BufferReader, File to get selected and get absolutepath for saving file notes and for opening already saved file notes. In other make a cut, copy, paste functions for the file’s notes.

- Step-7: After write a syntax function for showing message for all other JMenuItem’s like for Save when it is disclosed shows “Operation has been cancelled”, About Notes shows “Created by: S. Khowshikk Sharma” and after clicking Close the message box shows as “Logout Successful”.

- Step-8: The third panel contains the transaction details which are S.no, Account Number, Status and the Amount.

IMPLEMENTATION

Frame: 1

```
package PRJT_26;
import javax.imageio.ImageIO;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.awt.image.ImageObserver;
import java.io.File;
import java.io.*;
import java.io.IOException;
import javax.swing.plaf.metal.*;
import javax.swing.text.*;
public class M2_LOGIN extends Canvas implements ActionListener{
	private static final long serialVersionUID = 1L;
	private static JLabel label1,label2,label3;
	private static JTextField username;
	private static JButton button1,button2;
	private static JPasswordField password;
	private static ImageIcon img;

	public static void main(String args[]) throws IOException {
		JPanel panel=new JPanel();
		panel.setLayout(null);
		panel.setBackground(Color.white);
		JFrame frame=new JFrame() ;
		frame.setTitle("LOGIN PAGE");
```

```
        frame.setLocation(new Point(500,200));
        frame.add(panel);
        frame.setSize(new Dimension(500,400));
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        File file = new File("C:\\Users\\Hello\\eclipseworkspace\\Project_No_26\\12345678.jpg");
      BufferedImage bufferedImage = ImageIO.read(file);
    ImageIcon imageIcon = new ImageIcon(bufferedImage);
    label3=new JLabel(imageIcon);
        label3.setBounds(50,0,400,200);
        panel.add(label3);
        label1=new JLabel("<html><font color='black'>Username</font></html>");
        label1.setBackground(Color.BLUE);
        label1.setBounds(150,190,70,20);
        panel.add(label1);
        username = new JTextField();
        username.setBounds(150,209,193,28);
        panel.add(username);
        label2= new JLabel("<html><font color='black'>Password</font></html>");
        label2.setBounds(150,237,70,20);
        panel.add(label2);
        password=new JPasswordField();
        password.setBounds(150,256,193,28);
        panel.add(password);
        button1 = new JButton("Login");
        button1.setBounds(150,295,90,25);
        button1.setForeground(Color.WHITE);
        button1.setBackground(Color.BLACK);
        button1.addActionListener((ActionListener) new M2_LOGIN());
        panel.add(button1);
         button2 = new JButton("Reset");
         button2.setBounds(250,295,90,25);
         button2.setForeground(Color.WHITE);
```

```
                button2.setBackground(Color.BLACK);
                button2.addActionListener((ActionListener) new M2_LOGIN());
                panel.add(button2);
               frame.setVisible(true);
        }

        public void actionPerformed(ActionEvent e) {
               JFrame frame=new JFrame() ;
               frame.setLocation(new Point(500, 300));
               frame.setSize(new Dimension(400, 200));
               frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                 if(e.getSource()==button2) {
                   username.setText("");
                   password.setText("");
                  }
                  if(e.getSource()==button1) {
                   String u= username.getText();
                   String p = String.valueOf(password.getPassword());
                   if(u.equals("kluniversity")&&p.equals("java#15")) {
                           M3_NOTES notes=new M3_NOTES();
                           JOptionPane.showMessageDialog(frame,"Login Succsesful...");
                           username.setText("");
                           password.setText("");
                   }
                   else {
                           JOptionPane.showMessageDialog(null," Wrong Username or Password ");
                   }
                  }
        }
}
```

Frame: 2

```
package PRJT_26;

import java.awt.*;

import javax.swing.*;

import java.io.*;

import java.awt.event.*;

import javax.swing.plaf.metal.*;

import javax.swing.text.*;

public class M3_NOTES extends JFrame implements ActionListener {


        JFrame f = new JFrame("Notes");

   JTextArea t = new JTextArea();

   Font fnt =new Font("Arial",Font.PLAIN,15);

        Container con=new Container();

        JScrollPane sbr=new
JScrollPane(t,JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED,JScrollPane.HORIZONTAL_SCROLLBAR_
AS_NEEDED);

   M3_NOTES()

   {

      try {

         UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");

         MetalLookAndFeel.setCurrentTheme(new OceanTheme());

      }

      catch (Exception e) {

      }


      f.setLocation(new Point(300,100));

      f.setSize(new Dimension(900,600));

      t.setFont(fnt);

                  t.setLineWrap(true);

                  t.setWrapStyleWord(true);
```

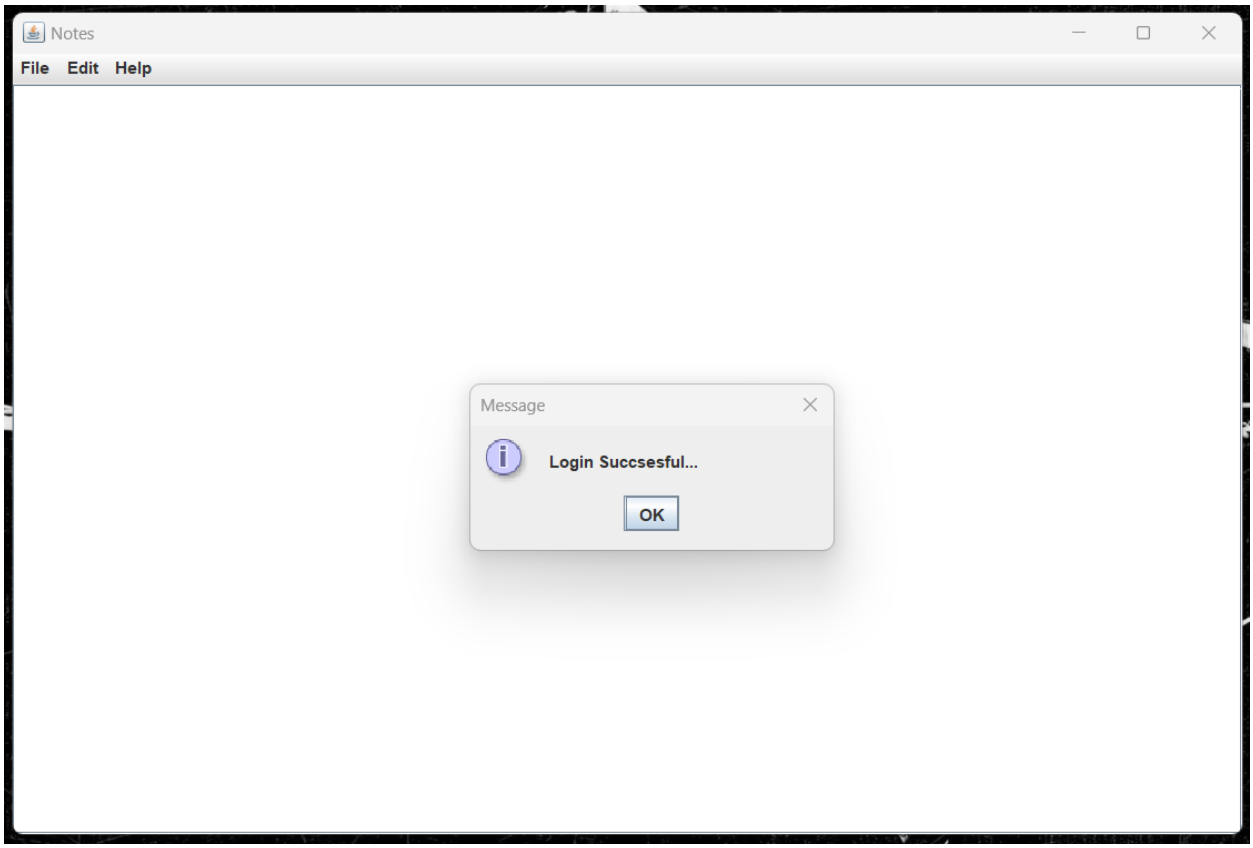
```
t.setVisible(true);
f.add(sbr);

JMenuBar mb = new JMenuBar();
JMenu m1 = new JMenu("File");
JMenuItem mi1 = new JMenuItem("New");
JMenuItem mi2 = new JMenuItem("Open");
JMenuItem mi3 = new JMenuItem("Save");
JMenuItem mi9 = new JMenuItem("Print");
mi1.addActionListener(this);
mi2.addActionListener(this);
mi3.addActionListener(this);
mi9.addActionListener(this);
m1.add(mi1);
m1.add(mi2);
m1.add(mi3);
m1.addSeparator();
m1.add(mi9);
JMenu m2 = new JMenu("Edit");
JMenuItem mi4 = new JMenuItem("Cut");
JMenuItem mi5 = new JMenuItem("Copy");
JMenuItem mi6 = new JMenuItem("Paste");
mi4.addActionListener(this);
mi5.addActionListener(this);
mi6.addActionListener(this);
m2.add(mi4);
m2.add(mi5);
m2.add(mi6);
JMenu m3 = new JMenu("Help");
JMenuItem mh=new JMenuItem("About Notes");
JMenuItem mc = new JMenuItem("Close");
mh.addActionListener(this);
```

```
    mc.addActionListener(this);
    m3.add(mh);
    m3.add(mc);
    mb.add(m1);
    mb.add(m2);
    mb.add(m3);
    f.setJMenuBar(mb);

    f.setVisible(true);
}

public void actionPerformed(ActionEvent e)
{
    String s=e.getActionCommand();
    if (s.equals("Cut")) {
        t.cut();
    }
    else if (s.equals("Copy")) {
        t.copy();
    }
    else if (s.equals("Paste")) {
        t.paste();
    }
    else if (s.equals("Save")) {
        JFileChooser j = new JFileChooser("f:");
        int r=j.showSaveDialog(null);
        if (r==JFileChooser.APPROVE_OPTION) {
            File fi =new File(j.getSelectedFile().getAbsolutePath());
            try {
                FileWriter wr=new FileWriter(fi,false);
                BufferedWriter w=new BufferedWriter(wr);
                w.write(t.getText());
```

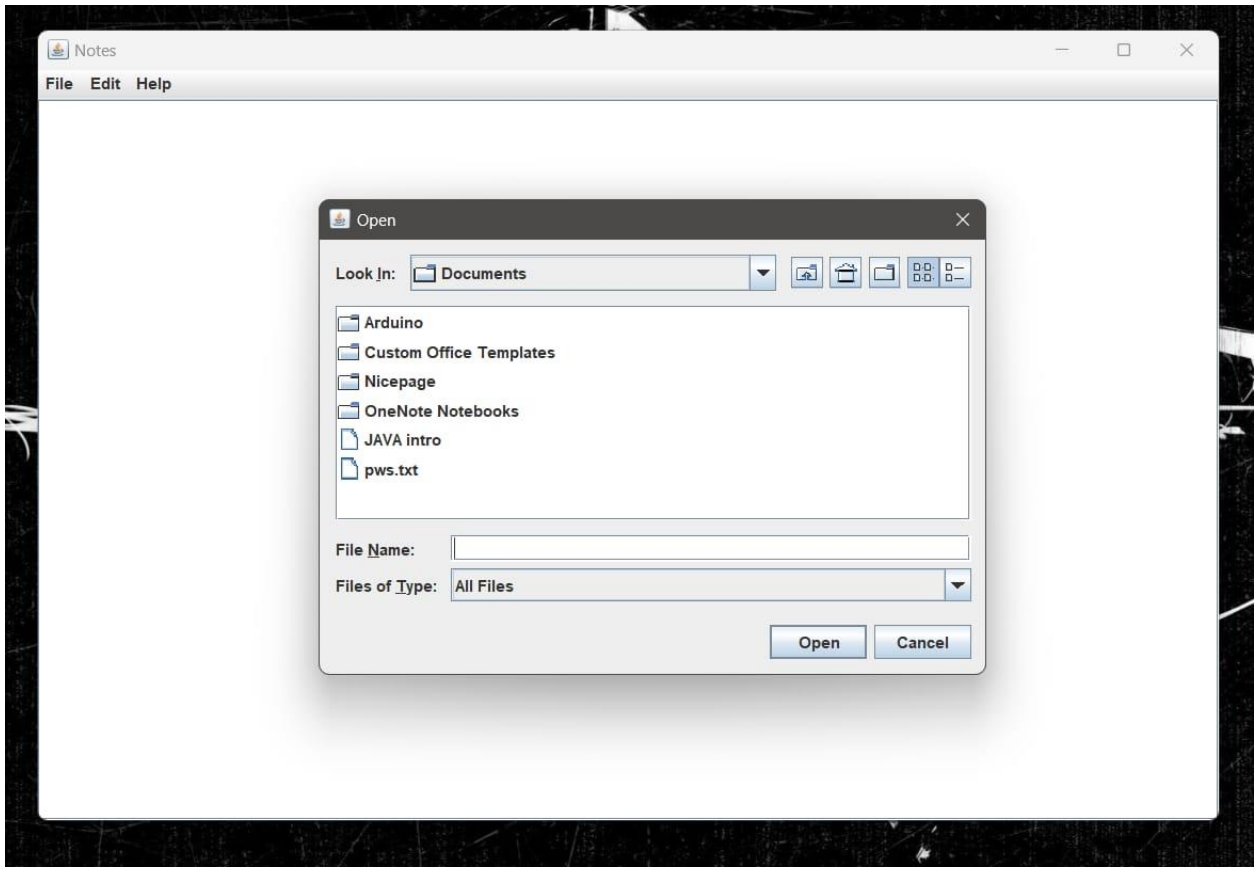
```
            w.flush();
            w.close();
        }
        catch (Exception evt) {
                JOptionPane.showMessageDialog(f, "The following operation is disabled");
        }
    }
    else
                JOptionPane.showMessageDialog(f, "The following operation is cancelled");
}
else if (s.equals("Print")) {
    try {
        t.print();
    }
    catch (Exception evt) {
                JOptionPane.showMessageDialog(f, "The following operation is disabled");
    }
}
else if (s.equals("Open")) {
    JFileChooser j = new JFileChooser("f:");
    int r=j.showOpenDialog(null);
    if (r==JFileChooser.APPROVE_OPTION) {
        File fi = new File(j.getSelectedFile().getAbsolutePath());

        try {
            String s1="",s2="";
            FileReader fr=new FileReader(fi);
            BufferedReader br=new BufferedReader(fr);
            s2=br.readLine();
            while ((s1=br.readLine())!= null) {
                s2=s2+"\n"+s1;
            }
```

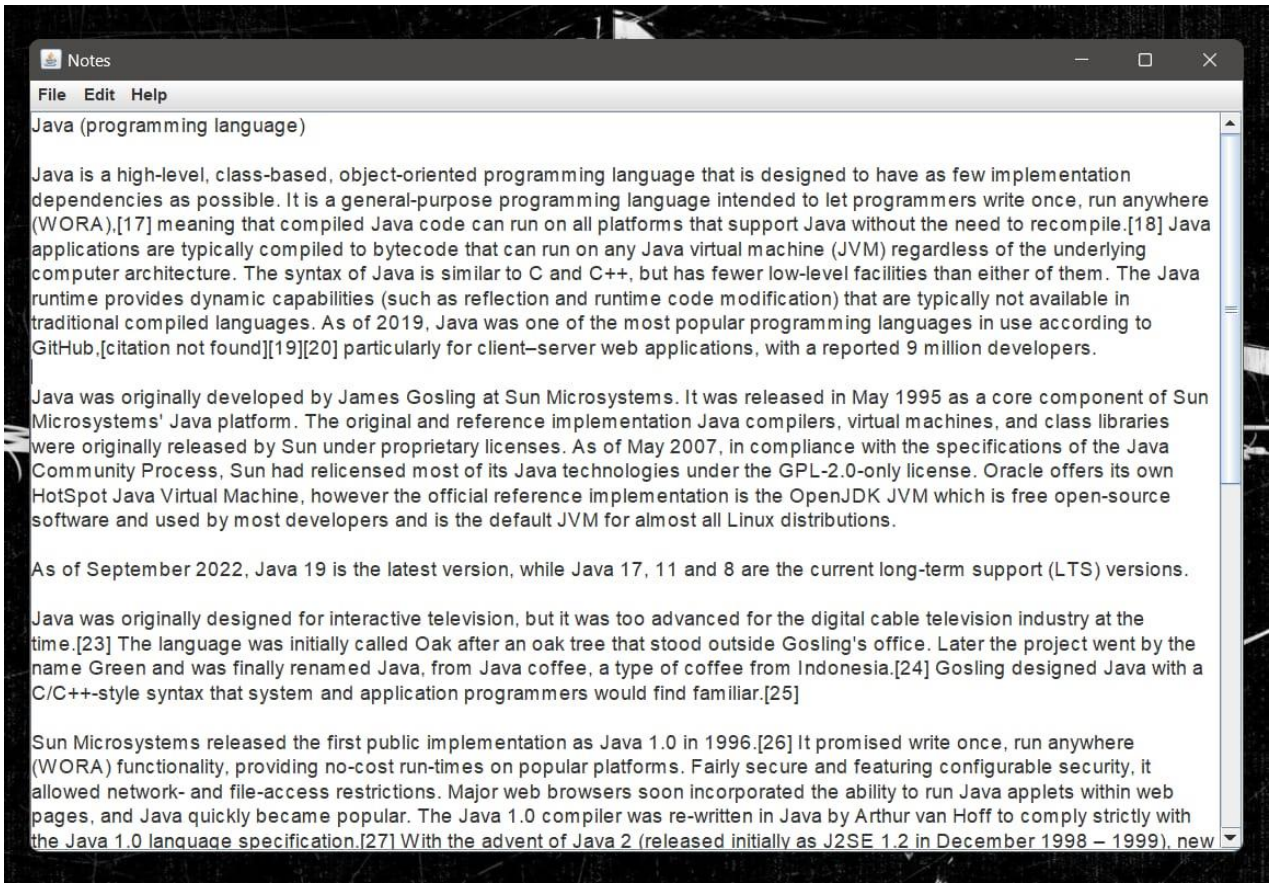
```
                t.setText(s2);
            }
            catch (Exception evt) {
                JOptionPane.showMessageDialog(f, "The following operation is disabled");
            }
        }
        else
            JOptionPane.showMessageDialog(f, "The following operation is cancelled");
    }
    else if (s.equals("New")) {
        t.setText("");
    }
    else if (s.equals("Close")) {
        f.setVisible(false);
        JOptionPane.showMessageDialog(f,"Logout Scuccessful");
    }
    else if (s.equals("About Notes")) {
       JOptionPane.showMessageDialog(f,"Created by : Khowshikk Sharma");
    }
  }
  public static void main(String args[])
  {
    M3_NOTES e = new M3_NOTES();
  }
}
```

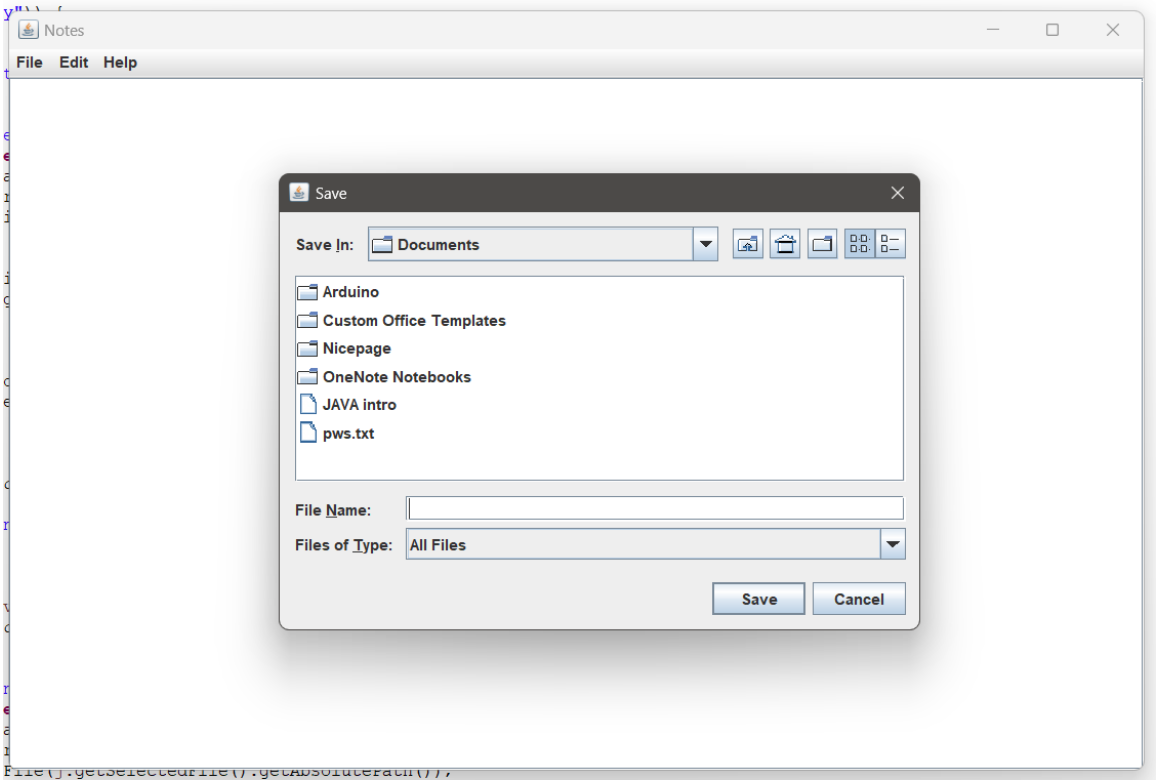
INTEGRATION & SYSTEM TESTING

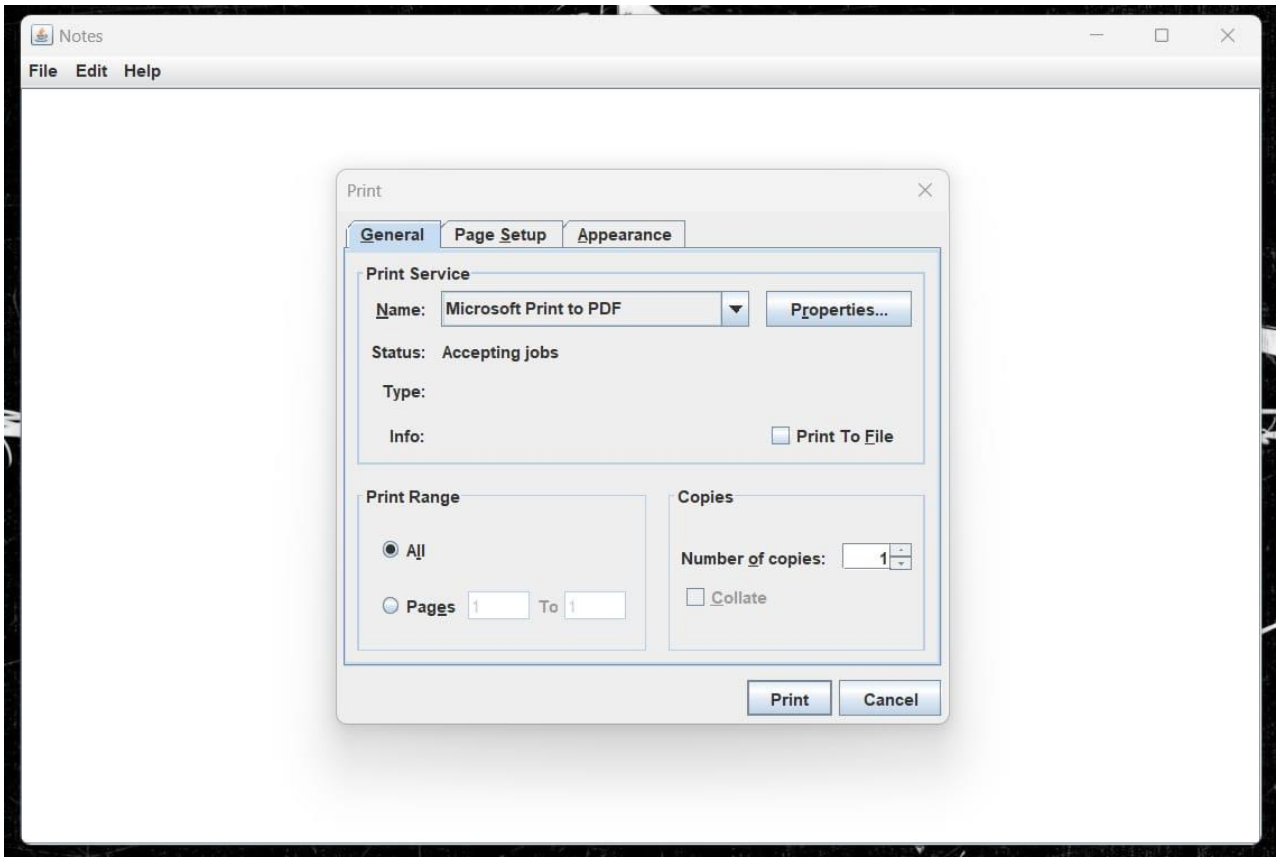


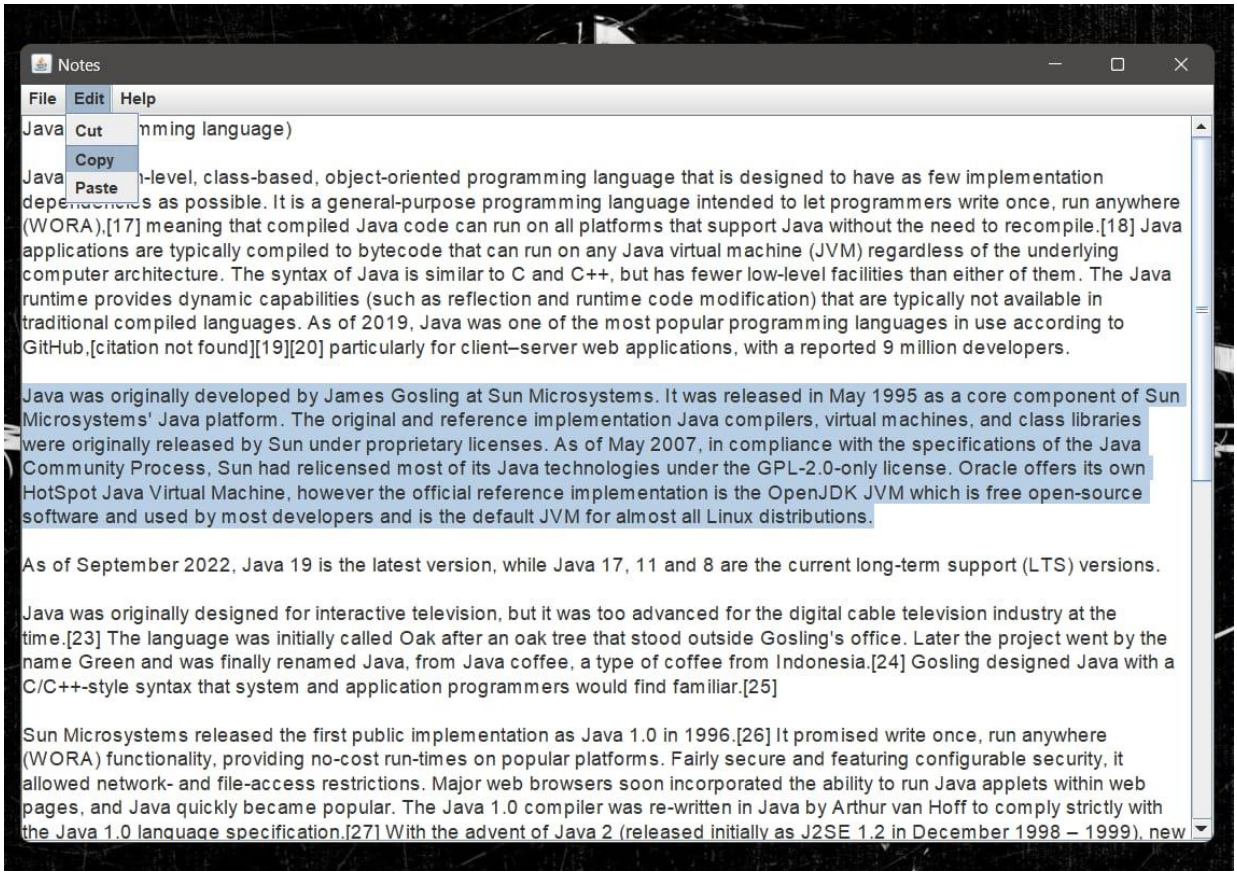

Notes
File
Edit
Help
New
Open
Save
Print

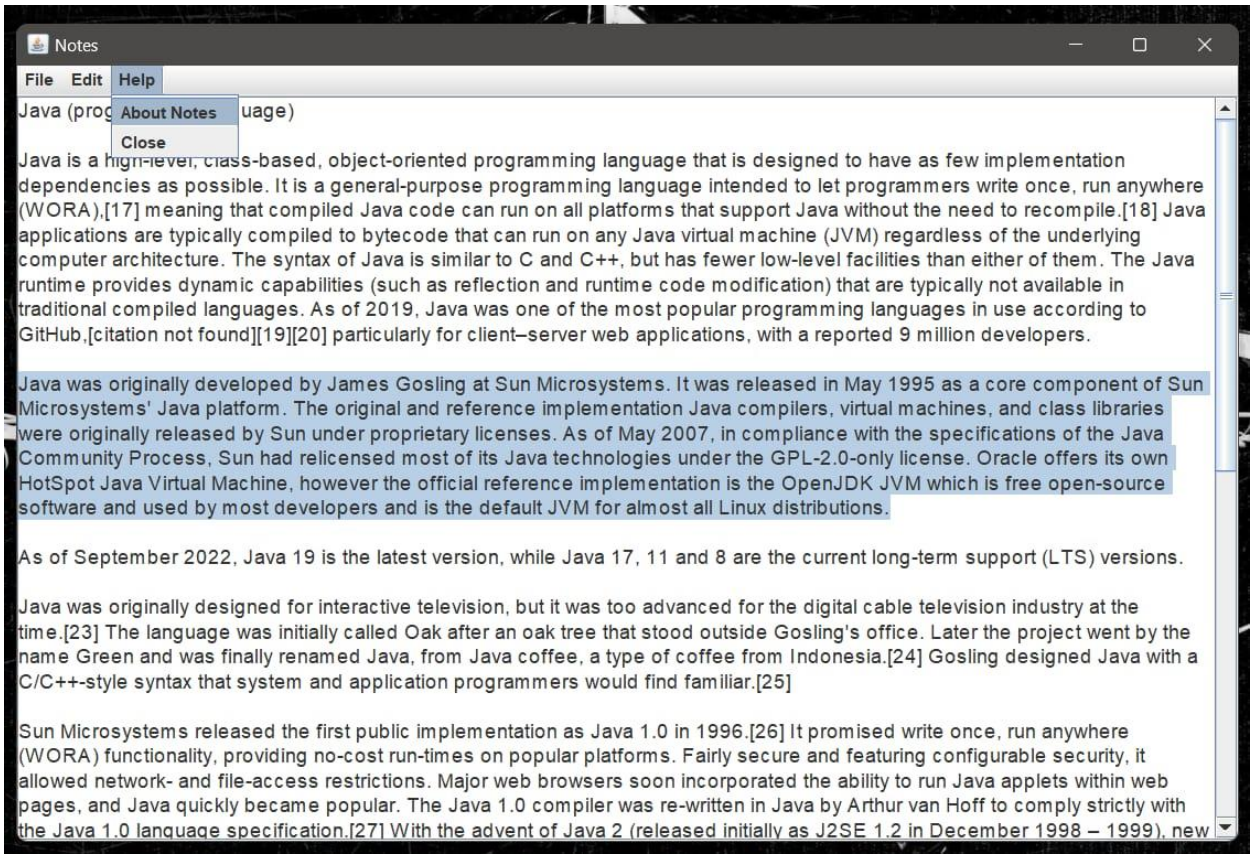

Notes
File
Edit
Help
Open
Look In:
Documents
Arduino
Custom Office Templates
Nicepage
OneNote Notebooks
JAVA intro
pws.txt
File Name:
Files of Type:
All Files
Open
Cancel

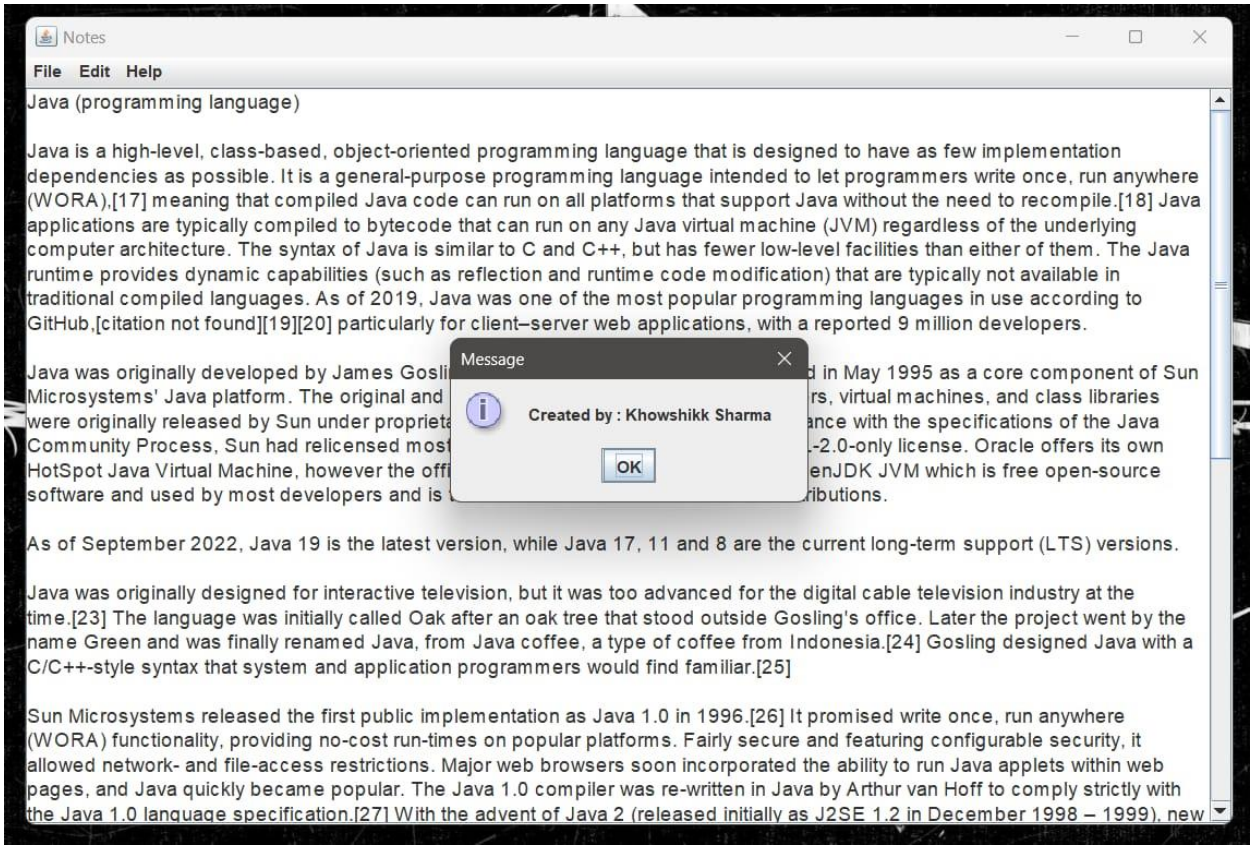

Notes
File Edit Help
Java (programming language)
Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA),[17] meaning that compiled Java code can run on all platforms that support Java without the need to recompile.[18] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub,[citation not found][19][20] particularly for client–server web applications, with a reported 9 million developers.
Java was originally developed by James Gosling at Sun Microsystems. It was released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.
As of September 2022, Java 19 is the latest version, while Java 17, 11 and 8 are the current long-term support (LTS) versions.
Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time.[23] The language was initially called Oak after an oak tree that stood outside Gosling's office. Later the project went by the name Green and was finally renamed Java, from Java coffee, a type of coffee from Indonesia.[24] Gosling designed Java with a C/C++-style syntax that system and application programmers would find familiar.[25]
Sun Microsystems released the first public implementation as Java 1.0 in 1996.[26] It promised write once, run anywhere (WORA) functionality, providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run Java applets within web pages, and Java quickly became popular. The Java 1.0 compiler was re-written in Java by Arthur van Hoff to comply strictly with the Java 1.0 language specification.[27] With the advent of Java 2 (released initially as J2SE 1.2 in December 1998 – 1999), new

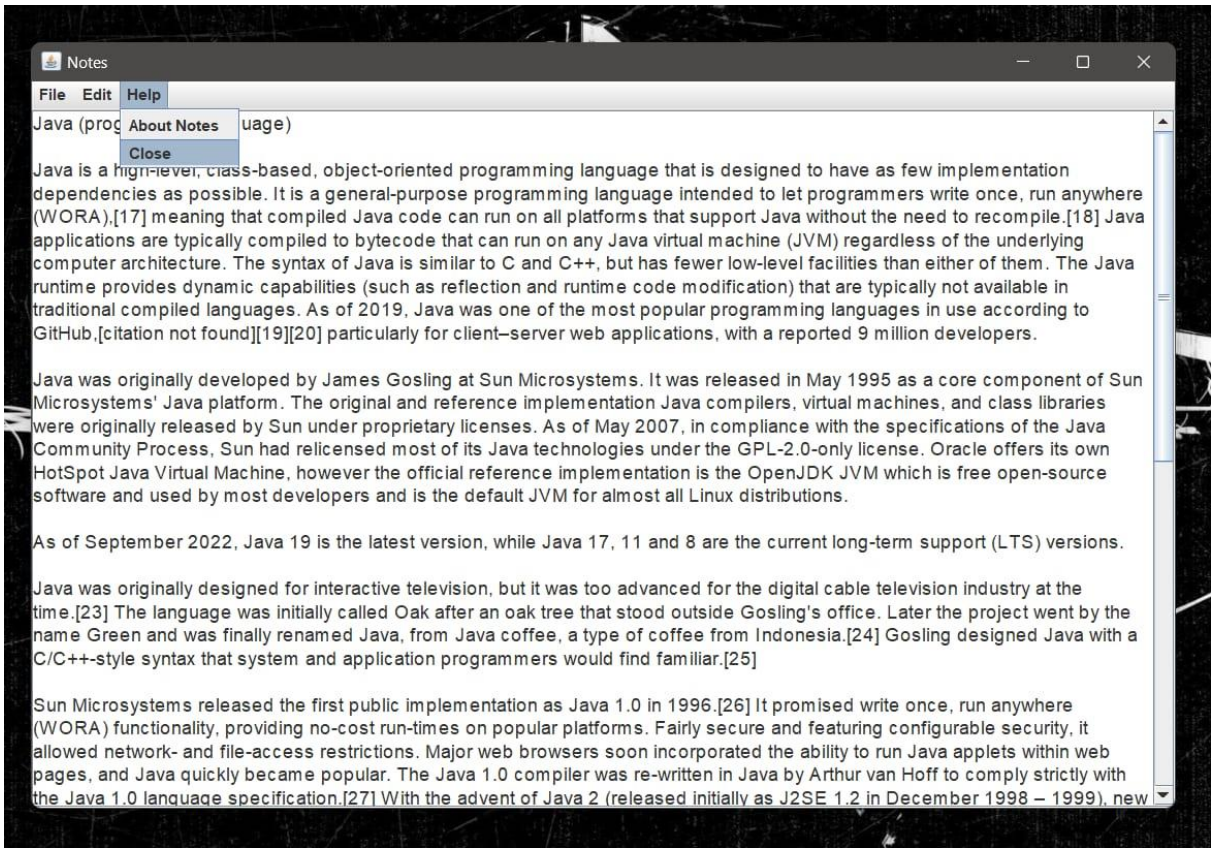

Notes
File Edit Help
Save
Save In: Documents
Arduino
Custom Office Templates
Nicepage
OneNote Notebooks
JAVA intro
pws.txt
File Name:
Files of Type: All Files
Save
Cancel

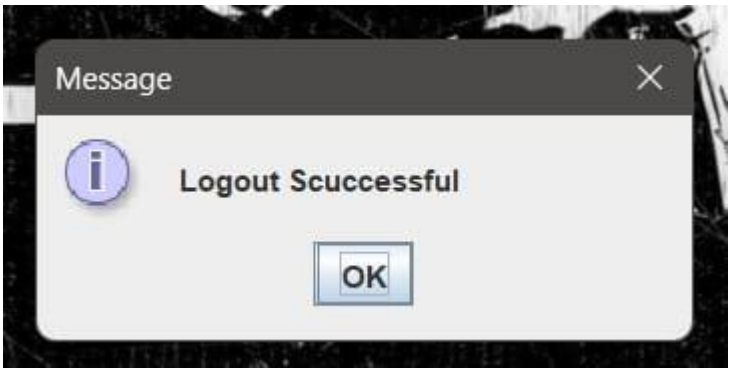

Notes
File Edit Help
Print
General Page Setup Appearance
Print Service
Name: Microsoft Print to PDF
Properties...
Status: Accepting jobs
Type:
Info:
Print To File
Print Range
All
Pages 1 To 1
Copies
Number of copies: 1
Collate
Print
Cancel


Notes
File Edit Help
Cut
Copy
Paste
mming language)
-level, class-based, object-oriented programming language that is designed to have as few implementation
s as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere
(WORA),[17] meaning that compiled Java code can run on all platforms that support Java without the need to recompile.[18] Java
applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying
computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java
runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in
traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to
GitHub,[citation not found][19][20] particularly for client–server web applications, with a reported 9 million developers.
Java was originally developed by James Gosling at Sun Microsystems. It was released in May 1995 as a core component of Sun
Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries
were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java
Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own
HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source
software and used by most developers and is the default JVM for almost all Linux distributions.
As of September 2022, Java 19 is the latest version, while Java 17, 11 and 8 are the current long-term support (LTS) versions.
Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the
time.[23] The language was initially called Oak after an oak tree that stood outside Gosling's office. Later the project went by the
name Green and was finally renamed Java, from Java coffee, a type of coffee from Indonesia.[24] Gosling designed Java with a
C/C++-style syntax that system and application programmers would find familiar.[25]
Sun Microsystems released the first public implementation as Java 1.0 in 1996.[26] It promised write once, run anywhere
(WORA) functionality, providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it
allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run Java applets within web
pages, and Java quickly became popular. The Java 1.0 compiler was re-written in Java by Arthur van Hoff to comply strictly with
the Java 1.0 language specification.[27] With the advent of Java 2 (released initially as J2SE 1.2 in December 1998 – 1999), new

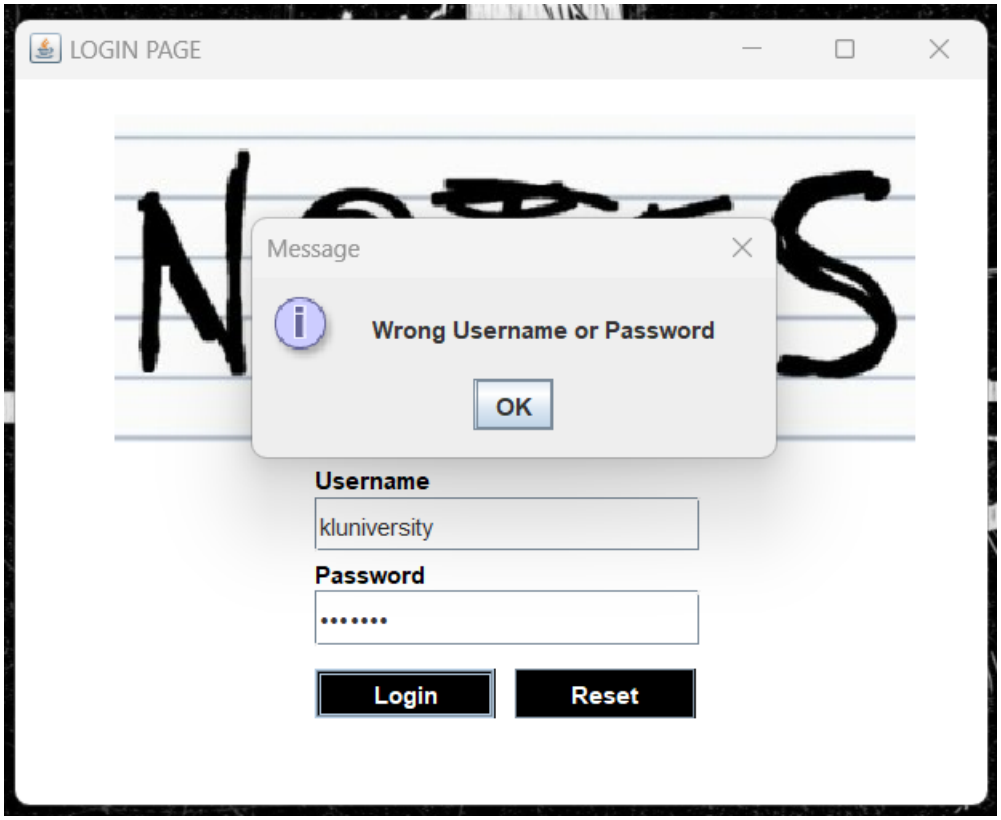

Notes
File Edit Help
About Notes
Close
Java (prog uage)
Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA),[17] meaning that compiled Java code can run on all platforms that support Java without the need to recompile.[18] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub,[citation not found][19][20] particularly for client–server web applications, with a reported 9 million developers.
Java was originally developed by James Gosling at Sun Microsystems. It was released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.
As of September 2022, Java 19 is the latest version, while Java 17, 11 and 8 are the current long-term support (LTS) versions.
Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time.[23] The language was initially called Oak after an oak tree that stood outside Gosling's office. Later the project went by the name Green and was finally renamed Java, from Java coffee, a type of coffee from Indonesia.[24] Gosling designed Java with a C/C++-style syntax that system and application programmers would find familiar.[25]
Sun Microsystems released the first public implementation as Java 1.0 in 1996.[26] It promised write once, run anywhere (WORA) functionality, providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run Java applets within web pages, and Java quickly became popular. The Java 1.0 compiler was re-written in Java by Arthur van Hoff to comply strictly with the Java 1.0 language specification.[27] With the advent of Java 2 (released initially as J2SE 1.2 in December 1998 – 1999), new


Notes
File Edit Help
Java (programming language)
Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA),[17] meaning that compiled Java code can run on all platforms that support Java without the need to recompile.[18] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub,[citation not found][19][20] particularly for client–server web applications, with a reported 9 million developers.
Message
Created by : Khowshikk Sharma
OK
Java was originally developed by James Gosli d in May 1995 as a core component of Sun Microsystems' Java platform. The original and rs, virtual machines, and class libraries were originally released by Sun under proprieta ance with the specifications of the Java Community Process, Sun had relicensed most L-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the offi enJDK JVM which is free open-source software and used by most developers and is t ributions.
As of September 2022, Java 19 is the latest version, while Java 17, 11 and 8 are the current long-term support (LTS) versions.
Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time.[23] The language was initially called Oak after an oak tree that stood outside Gosling's office. Later the project went by the name Green and was finally renamed Java, from Java coffee, a type of coffee from Indonesia.[24] Gosling designed Java with a C/C++-style syntax that system and application programmers would find familiar.[25]
Sun Microsystems released the first public implementation as Java 1.0 in 1996.[26] It promised write once, run anywhere (WORA) functionality, providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run Java applets within web pages, and Java quickly became popular. The Java 1.0 compiler was re-written in Java by Arthur van Hoff to comply strictly with the Java 1.0 language specification.[27] With the advent of Java 2 (released initially as J2SE 1.2 in December 1998 – 1999), new


Notes
File
Edit
Help
About Notes
Close
Java (prog uage)
Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA),[17] meaning that compiled Java code can run on all platforms that support Java without the need to recompile.[18] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub,[citation not found][19][20] particularly for client–server web applications, with a reported 9 million developers.
Java was originally developed by James Gosling at Sun Microsystems. It was released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.
As of September 2022, Java 19 is the latest version, while Java 17, 11 and 8 are the current long-term support (LTS) versions.
Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time.[23] The language was initially called Oak after an oak tree that stood outside Gosling's office. Later the project went by the name Green and was finally renamed Java, from Java coffee, a type of coffee from Indonesia.[24] Gosling designed Java with a C/C++-style syntax that system and application programmers would find familiar.[25]
Sun Microsystems released the first public implementation as Java 1.0 in 1996.[26] It promised write once, run anywhere (WORA) functionality, providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run Java applets within web pages, and Java quickly became popular. The Java 1.0 compiler was re-written in Java by Arthur van Hoff to comply strictly with the Java 1.0 language specification.[27] With the advent of Java 2 (released initially as J2SE 1.2 in December 1998 – 1999), new


Message
Logout Scuccessful
OK


LOGIN PAGE
NOTES
Username
Password
Login
Reset


LOGIN PAGE
Message
Wrong Username or Password
OK
Username
kluniversity
Password
Login
Reset

CONCLUSION

Java program presented here is a simple and user-friendly text editor application that provides basic functionality for creating, opening, saving, and printing text documents. The program uses various Java libraries to create a graphical user interface with menus and buttons to perform different operations on the text area. The program also handles errors and exceptions gracefully and provides appropriate messages to the user.

The application is easy to use, even for those with little or no programming experience, and provides a handy tool for creating and editing text documents. The use of a JMenuBar and menu items make it easy to access and perform various operations on the text area.

Overall, this Java program is an excellent example of how to create a basic text editor application in Java using various libraries and GUI components. The code is well-structured, easy to read, and provides a solid foundation for further development and customization. The program's simplicity and user-friendliness make it an ideal tool for beginners who want to learn Java programming and GUI development. So, using the concepts of GUI.

By completing this project, we have learnt how to:

- Create Frames in Java.
- Add colors, panels, labels, text fields, buttons and images to the Frames with Files.
- Add borders to the panels, images, test area, scrollbar and File layouts
- Implement Event actions on buttons.