TRINITY INTERNATIONAL COLLEGE

(Tribhuvan University Affiliated)

Lab Assignment 2, 3: Advance Java Programming

Submitted By: **Submitted to:**

Name:

Program: **B. Sc. (CSIT)**

Roll No:

Semester: seventh (7th)

Date:

KATHMANDU, NEPAL

2020

Unit 2, 3: User Interface Components with Swing, Event Handling

1) Write a program using components to add two numbers. Use text fields. For inputs and output. Your program should display the result when the user presses a button. [2069]

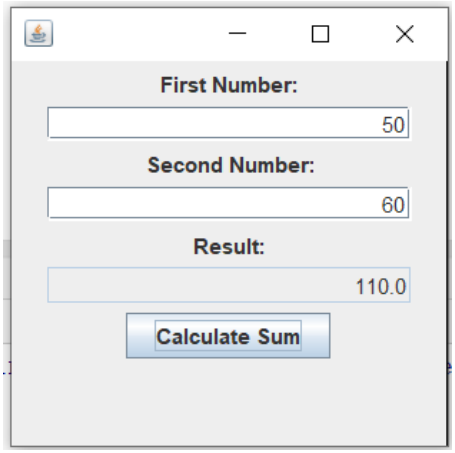
⇨

Program

```
package Q01_AddTwoNumbers;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class AddTwoNumbers extends JFrame
{
    public static void main(String[] args)
    {
        AddTwoNumbers frame = new AddTwoNumbers();
        frame.setVisible(true);
        frame.setBounds(500,100,250,250);

    }
    public AddTwoNumbers()
    {
        setLayout(new FlowLayout());
        JLabel firstTextFieldLabel = new JLabel("First
                                                Number:");
        JTextField firstTextField = new JTextField(20);
        add(firstTextFieldLabel);
        add(firstTextField);
        firstTextField.setComponentOrientation
                    (ComponentOrientation.RIGHT_TO_LEFT);

        JLabel secondTextLabel = new JLabel("Second
                                                Number:");
        JTextField secondTextField = new JTextField(20);
        add(secondTextLabel);
        add(secondTextField);

        secondTextField.setComponentOrientation
                    (ComponentOrientation.RIGHT_TO_LEFT);

        JLabel displayResultLabel = new
                                JLabel("Result:");
        JTextField displayResultField = new
                                    JTextField(20);
        add(displayResultLabel);
        add(displayResultField);
        displayResultField.setEditable(false);
```

```
        displayResultField.setComponentOrientation
                (ComponentOrientation.RIGHT_TO_LEFT);

    JButton calculateSum = new JButton("Calculate Sum");
    add(calculateSum);

     calculateSum.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                double firstNumber = Double.parseDouble
                            (firstTextField.getText());

                double secondNumber = Double.parseDouble
                           (secondTextField.getText());

                double sum = firstNumber + secondNumber;
                displayResultField.setText
                                (String.valueOf(sum));
            }
        });
        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

Output:



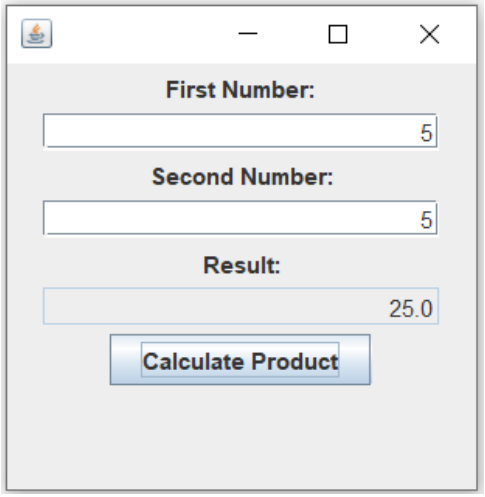
2) Write a program using swing components to multiply two numbers. Use text fields for inputs and output. Your program should display the result when the user presses a button. [2070]
⇨

Program

```
package Q02_MultiplyTwoNumbers;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MultiplyTwoNumbers extends JFrame
{
    public static void main(String[] args)
    {
        MultiplyTwoNumbers frame = new
                                    MultiplyTwoNumbers();
        frame.setVisible(true);
        frame.setBounds(500,100,250,250);

    }
    public MultiplyTwoNumbers()
    {
        setLayout(new FlowLayout());
        JLabel firstTextFieldLabel = new JLabel("First
                                                Number:");
        JTextField firstTextField = new JTextField(20);
        add(firstTextFieldLabel);
        add(firstTextField);
        firstTextField.setComponentOrientation
                    (ComponentOrientation.RIGHT_TO_LEFT);
        JLabel secondTextLabel = new JLabel("Second
                                             Number:");
        JTextField secondTextField = new JTextField(20);
        add(secondTextLabel);
        add(secondTextField);

        secondTextField.setComponentOrientation
                    (ComponentOrientation.RIGHT_TO_LEFT);

        JLabel displayResultLabel =new JLabel("Result:");
        JTextField displayResultField =new JTextField(20;
        add(displayResultLabel);
        add(displayResultField);
        displayResultField.setEditable(false);
        displayResultField.setComponentOrientation
                    (ComponentOrientation.RIGHT_TO_LEFT);
        JButton calculateProduct = new JButton("Calculate
                                              Product");
        add(calculateProduct);
```

```
        calculateProduct.addActionListener(new
                                        ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                double firstNumber = Double.parseDouble
                                (firstTextField.getText());

                double secondNumber = Double.parseDouble
                                (secondTextField.getText());

                double sum = firstNumber * secondNumber;
                displayResultField.setText
                                (String.valueOf(sum));
            }
        });
        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

Output



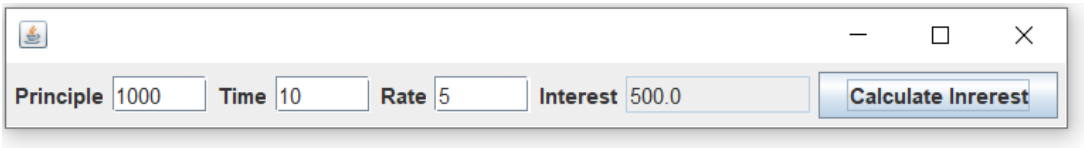
3) Write a program using swing components to find simple interest. Use text fields for inputs and output. Your program should display the result when the user presses a button. [2071, 2074]
⇨

Program

```
package Q03_SimpleInterest;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SimpleInterest extends JFrame
{
    public static void main(String[] args)
    {
        SimpleInterest frame = new SimpleInterest();
        frame.setVisible(true);

    }

    public SimpleInterest()
    {

        //place your code here
        setLayout(new FlowLayout());


        add(new JLabel("Principle"));
        JTextField principleTextField =new JTextField(5);
        add(principleTextField);

        add(new JLabel("Time"));
        JTextField timeTextField = new JTextField(5);
        add(timeTextField);

        add(new JLabel("Rate"));
        JTextField rateTextField = new JTextField(5);
        add(rateTextField);


        add(new JLabel("Interest"));
        JTextField interestTextField =new JTextField(10);
        interestTextField.setEditable(false);
        add(interestTextField);

        JButton calculateInterest =new JButton("Calculate
                                                Inrerest");
        add(calculateInterest);
```

```
        calculateInterest.addActionListener(new
                                        ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                double principle = Double.parseDouble
                          (principleTextField.getText());

                double rate = Double.parseDouble
                                (rateTextField.getText());

                double time = Double.parseDouble
                                (timeTextField.getText());

                double calculateInterest =
                                (principle*rate*time)/100;

                interestTextField.setText(String.valueOf
                                    (calculateInterest));

            }
        });

        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

Output:



4) Design a GUI form using swing with a text field, a text label for displaying the input message "Input any string", and three buttons with caption "Check Palindrome", "Reverse", "Find Vowels". Write a complete program for above scenario and for checking palindrome in first button, reverse it after clicking second button and extract the vowels from it after clicking third button. [2075]

⇨

Program

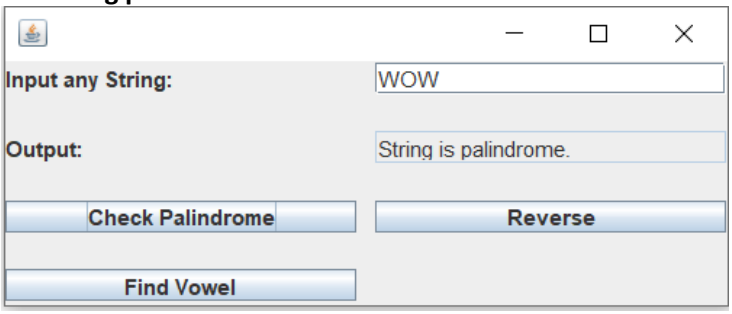
```
package Q04_PalindromeProgram;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Palindrome extends JFrame
{
    public static void main(String[] args)
    {
        Palindrome frame = new Palindrome();
        frame.setVisible(true);
    }
    public Palindrome()
    {
        setLayout(new GridLayout(4,1,10,20));


        JLabel inputLabel = new JLabel("Input any
                                        String: ");
        JTextField inputTextField = new JTextField(20);
        add(inputLabel);
        add(inputTextField);

        JLabel outputLabel = new JLabel("Output: ");
        JTextField outputTextField = new JTextField(20);
        add(outputLabel);
        add(outputTextField);
        outputTextField.setEditable(false);

        JButton checkPalindromeButton = new
        JButton("Check Palindrome");
        add(checkPalindromeButton);

        JButton reverseButton = new JButton("Reverse");
        add(reverseButton);

        JButton findVowelButton = new JButton("Find
                                                Vowel");
        add(findVowelButton);




        checkPalindromeButton.addActionListener(new
```

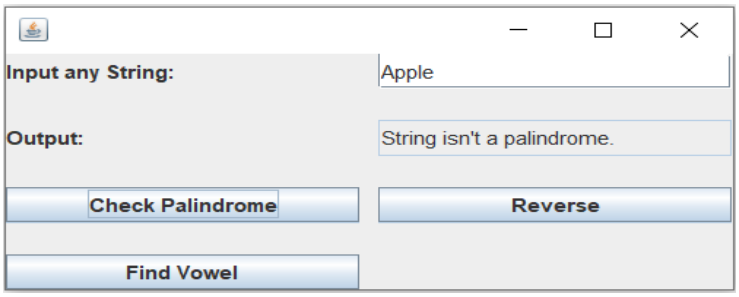
```
                                    ActionListener()
        {
           @Override
           public void actionPerformed(ActionEvent e)
            {
                String copyUserInput="";
                String userInput =inputTextField.
                                        getText();
                int length = userInput.length();

                for (int i = length-1; i>=0; i-- )
                {
                    copyUserInput = copyUserInput +
                                    userInput.charAt(i);
                }
                if (copyUserInput.equalsIgnoreCase
                                        (userInput))
                {
                    outputTextField.setText("String is
                                        palindrome.");
                }
                else                    {
                    outputTextField.setText("String
                                    isn't a palindrome.");
                }
            }
        });

        reverseButton.addActionListener(new
                                    ActionListener()
        {
          @Override
          public void actionPerformed(ActionEvent e)
            {
                String reverseUserInput="";
                String userInput = inputTextField.
                                        getText();
                int length = userInput.length();

                for (int i = length-1; i>=0; i-- )
                {
                   reverseUserInput = reverseUserInput +
                                    userInput.charAt(i);
                }

          outputTextField.setText("Reverse String is: "+
                                    reverseUserInput);
            }
        });





        findVowelButton.addActionListener(new
```

```
                                        ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                char[] vowel={'a','e','i','o','u',
                              'A','E','I','O','U'};
                String userInput = inputTextField.
                                        getText();
                int length = userInput.length();
                char[] extractedVowel= new char[length];
                String showVowel="";


                for (int i =0; i<=length-1; i++ )
                {
                    for (int j = 0; j<=vowel.length-1;
                                            j++)
                    {
                        if(userInput.charAt(i)==
                                        vowel[j])
                        {
                            extractedVowel[i] =
                                    userInput.charAt(i);

                            showVowel = showVowel +
                            String.valueOf
                                    (extractedVowel[i]);
                        }
                    }
                }

                 outputTextField.setText
                                ("Vowels: "+showVowel);
            }
        });

        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

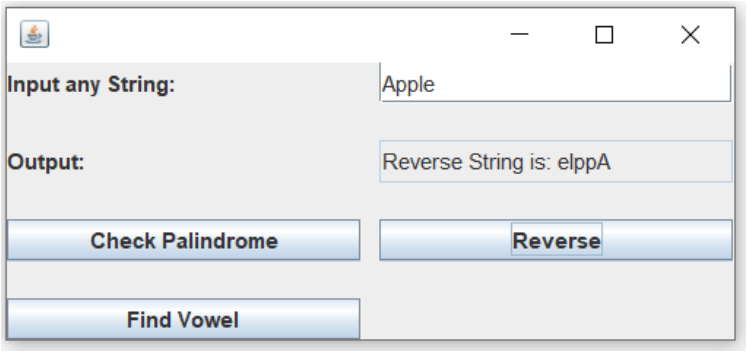
<u>output</u>

Checking palindrome or not

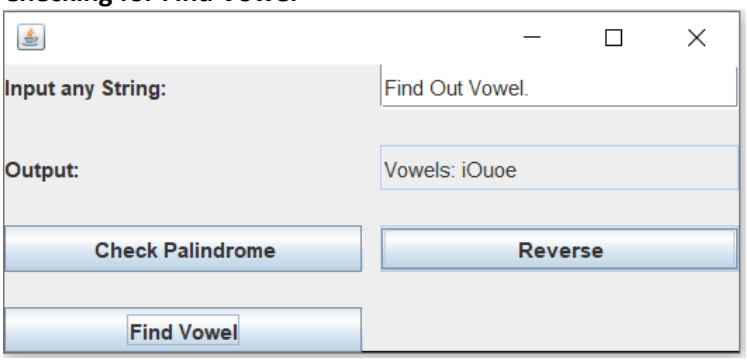




Checking for Reverse



Checking for Find Vowel

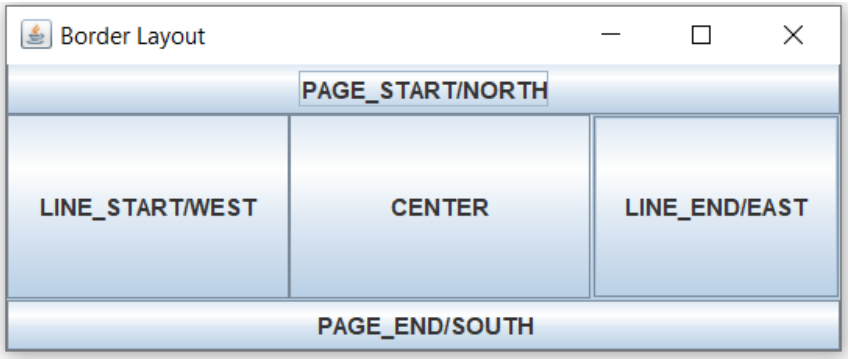


5) Write a program to illustrate the use of BorderLayout. [2073]
⇨

Program

```
package Q05_BorderLayout;

import javax.swing.*;
import java.awt.*;

public class BorderLayoutDemo extends JFrame
{
    public static void main(String[] args)
    {
        BorderLayoutDemo frame = new BorderLayoutDemo();
        frame.setVisible(true);
        frame.setTitle("Border Layout");
    }


    public BorderLayoutDemo()
    {
        setLayout(new BorderLayout());

        JButton topButton = new JButton("PAGE_START/NORTH");
        JButton bottomButton = new JButton("PAGE_END/SOUTH");
        JButton leftButton = new JButton("LINE_START/WEST");
        JButton rightButton = new JButton("LINE_END/EAST");
        JButton centerButton = new JButton("CENTER");

        add(topButton,BorderLayout.PAGE_START);
        add(bottomButton,BorderLayout.PAGE_END);
        add(leftButton,BorderLayout.LINE_START);
        add(rightButton,BorderLayout.LINE_END);
        add(centerButton,BorderLayout.CENTER);

        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

output:



6) Write a program to calculate simple interest using
   a) FlowLayout
   b) GridLayout
   c) GridBagLayout

⇨

a) FlowLayout

Program

```
package Q06_a_SimpleInterestFlowLayout;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SimpleInterestFlowLayout extends JFrame
{
    public static void main(String[] args)
    {
        SimpleInterestFlowLayout frame = new
                                SimpleInterestFlowLayout();
        frame.setVisible(true);
    }
    public SimpleInterestFlowLayout()
    {
        setLayout(new FlowLayout());

        add(new JLabel("Principle"));
        JTextField principleTextField = new JTextField(5);
        add(principleTextField);

        add(new JLabel("Time"));
        JTextField timeTextField = new JTextField(5);
        add(timeTextField);

        add(new JLabel("Rate"));
        JTextField rateTextField = new JTextField(5);
        add(rateTextField);

        add(new JLabel("Interest"));
        JTextField interestTextField = new JTextField(10);
        interestTextField.setEditable(false);
        add(interestTextField);

        JButton calculateInterest = new JButton("Calculate
                                                Inrerest");
        add(calculateInterest);
```

```
        calculateInterest.addActionListener(new
                                        ActionListener()
        {

            @Override
            public void actionPerformed(ActionEvent e)
            {
                double principle = Double.parseDouble
                            (principleTextField.getText());
                double rate = Double.parseDouble
                                (rateTextField.getText());
                double time = Double.parseDouble
                                (timeTextField.getText());
                double calculateInterest =
                                (principle*rate*time)/100;

                interestTextField.setText
                      (String.valueOf(calculateInterest));

            }
        });

        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

Output

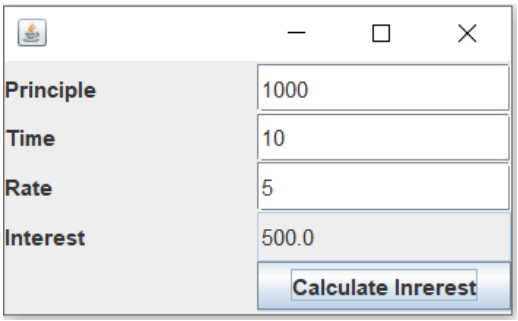
b) GridLayout

Program

```
package Q06_b_SimpleInterestGridLayout;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

    public class SimpleInterestGridLayout extends JFrame
{
        public static void main(String[] args)
        {
            SimpleInterestGridLayout app = new
                                SimpleInterestGridLayout();
            app.setVisible(true);
        }

        public SimpleInterestGridLayout()
        {

            //place your code here
            setLayout(new GridLayout(5,1));


            add(new JLabel("Principle"));
            JTextField principleTextField = new
                                        JTextField(5);
            add(principleTextField);

            add(new JLabel("Time"));
            JTextField timeTextField = new JTextField(5);
            add(timeTextField);

            add(new JLabel("Rate"));
            JTextField rateTextField = new JTextField(5);
            add(rateTextField);


            add(new JLabel("Interest"));
            JTextField interestTextField = new
                                       JTextField(10);
            interestTextField.setEditable(false);
            add(interestTextField);
            add(new JLabel(""));

            JButton calculateInterest = new
           JButton("Calculate Inrerest");
            add(calculateInterest);
```

```
            calculateInterest.addActionListener(new
                                        ActionListener()
            {
                @Override
               public void actionPerformed(ActionEvent e)
                {
                    double principle = Double.parseDouble
                          (principleTextField.getText());
                    double rate = Double.parseDouble
                              (rateTextField.getText());
                    double time = Double.parseDouble
                              (timeTextField.getText());
                    double calculateInterest =
                              (principle*rate*time)/100;

                    interestTextField.setText
                     (String.valueOf(calculateInterest));
                }
            });
            pack();
            setDefaultCloseOperation(EXIT_ON_CLOSE);
        }
    }
```

Output



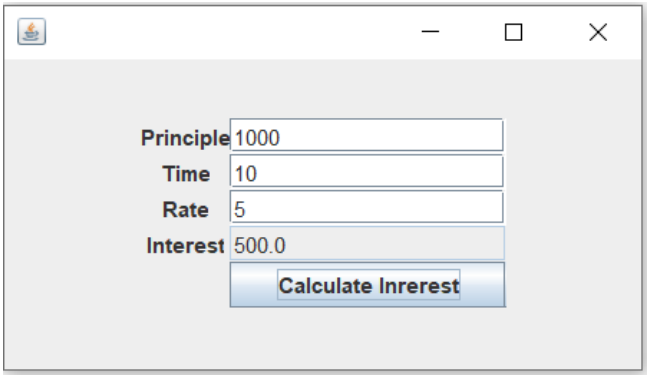
c) GridBagLayout

Program

```
package Q06_c_SimpleInterestGridBagLayout;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SimpleInterestGridBagLayout extends JFrame
{
    public static void main(String[] args)
    {
        SimpleInterestGridBagLayout app = new
                                SimpleInterestGridBagLayout();
       app.setVisible(true);
    }

    public SimpleInterestGridBagLayout()
    {
        setLayout(new GridBagLayout());
        GridBagConstraints s = new GridBagConstraints();

        //For Principle
        s.gridx =0;
        s.gridy =0;
        add(new JLabel("Principle"));
        JTextField principleTextField=new JTextField(15);
        add(principleTextField);

        //For Rate
        s.gridx =0;
        s.gridy =1;
        add(new JLabel("Time"),s);
        JTextField timeTextField = new JTextField(15);
        s.gridx = 1;
        add(timeTextField,s);

        //For Time
        s.gridx =0;
        s.gridy =2;
        add(new JLabel("Rate"),s);
        JTextField rateTextField = new JTextField(15);
        s.gridx=1;
        add(rateTextField,s);

        //For Interest Displaying Field
        s.gridx =0;
        s.gridy =3;
        add(new JLabel("Interest"),s);
        JTextField interestTextField =new JTextField(15);
        interestTextField.setEditable(false);
        s.gridx =1;
        add(interestTextField,s);
```

```
        //For Button
        s.fill =GridBagConstraints.BOTH;
        s.gridx =1;
        s.gridy =5;
        s.gridwidth= 1;
        JButton calculateInterest =new JButton("Calculate
                                                Inrerest");
        add(calculateInterest,s);

        calculateInterest.addActionListener(new
                                        ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                double principle = Double.parseDouble
                          (principleTextField.getText());

                double rate = Double.parseDouble
                               (rateTextField.getText());

                double time = Double.parseDouble
                               (timeTextField.getText());

                double calculateInterest =
                               (principle*rate*time)/100;

                interestTextField.setText
                     (String.valueOf(calculateInterest));
            }
        });

        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

Output:



7) Create a login form with username and password fields. Print “access granted” if the username and password both are “admin”, when user clicks on Login button. If authentication fails, print “access denied”.

⇨

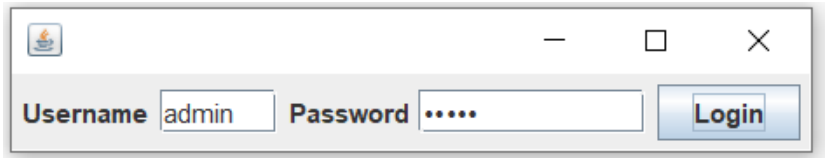
Program

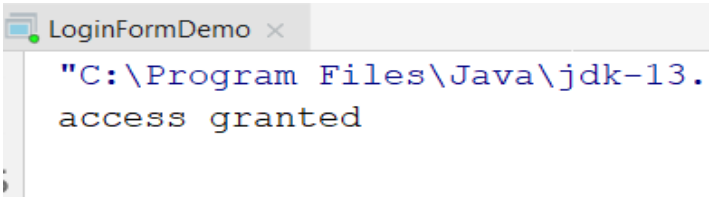
```
package Q07_LoginForm;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Arrays;

public class LoginFormDemo extends JFrame
{
    public static void main(String[] args)
    {
        LoginFormDemo app   =new LoginFormDemo();
        app.setVisible(true);
    }

    public LoginFormDemo ()
    {
        setLayout(new FlowLayout());
        add(new JLabel("Username"));
        JTextField LoginTextField = new JTextField(5);
        add(LoginTextField);

        add(new JLabel("Password"));
        JPasswordField LoginPasswordField = new
                                        JPasswordField(10);
        add(LoginPasswordField);

        JButton LoginButton = new JButton("Login");
        add(LoginButton);

        LoginButton.addActionListener(new ActionListener()
         {
            @Override
           public void actionPerformed(ActionEvent e)
            {
                String username = LoginTextField.getText();
                char[] password = LoginPasswordField
                                        .getPassword();
                char[] actualPassword =
                                {'a','d','m','i','n'};

                if (username.equals("admin") && Arrays.
                          equals(actualPassword,password))
                {
                    System.out.println("access granted");
                }
                else                    {
                    System.out.println("access denied");
                }
```

```
            }
        });
        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);

    }
}
```

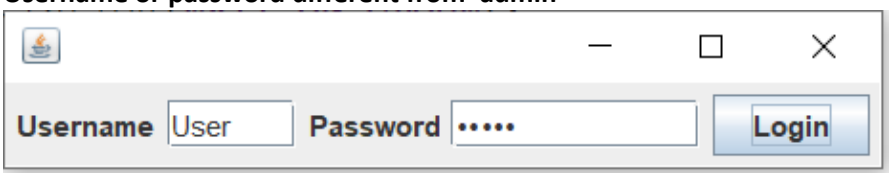
Output

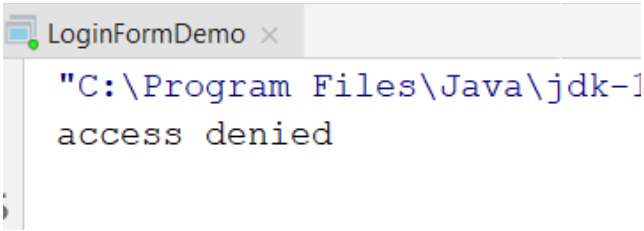
Username : admin
Password: admin





Username or password different from 'admin'





8) *(Optional)* Create a basic notepad app with the following features:
   a) New
   b) Open
   c) Save
   d) Exit
   Use JButton components to implement these features

9) Create an application with UI similar to the windows notepad app.

⇨

Program

```
package Q08_Q09_NotepadApp;

import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;

public class NotePadApp extends JFrame
{
    public static void main(String[] args)
    {

        NotePadApp frame = new NotePadApp();
        frame.setVisible(true);
        frame.setBounds(500,100,1000,500);
        frame.setTitle("JMenu Introduction");

    }
    JTextArea textArea;
    public NotePadApp()
    {
        Container c = this.getContentPane();
        c.setLayout(null);

        textArea = new JTextArea();
        textArea.setBounds(0,20,1000,500);
        c.add(textArea);
//********************** Creating Menu Bar ****************
        JMenuBar menuBar = new JMenuBar();
        menuBar.setBounds(0,0,1000,20);
        c.add(menuBar);

//***************** Adding Menus to the Menu Bar************
        //Adding  'File' Menu to menuBar.
        JMenu fileMenu = new JMenu("File");
        menuBar.add(fileMenu);



        //Adding  'Edit' Menu to Menu Bar.
```

```
        JMenu editMenu = new JMenu("Edit");
        menuBar.add(editMenu);

        //Adding  'Format' Menu to Menu Bar.
        JMenu formatMenu = new JMenu("Format");
        menuBar.add(formatMenu);

        //Adding  'View' Menu to Menu Bar.
        JMenu viewMenu = new JMenu("View");
        menuBar.add(viewMenu);

        //Adding  'View' Menu to Menu Bar.
        JMenu helpMenu = new JMenu("Help");
        menuBar.add(helpMenu);

//****************** Adding MenuItems to the Menu**********
        //Adding MenuItems to the 'File' Menu
        JMenuItem newMenuItem = new JMenuItem("New");
        fileMenu.add(newMenuItem);

        JMenuItem openMenuItem = new JMenuItem("Open...");
        fileMenu.add(openMenuItem);

        JMenuItem saveMenuItem = new JMenuItem("Save" );
        fileMenu.add(saveMenuItem);

        JMenuItem saveAsMenuItem = new JMenuItem("Save
                                                  AS...");
        fileMenu.add(saveAsMenuItem);

        fileMenu.addSeparator();

        JMenuItem pageSetupMenuItem = new JMenuItem("Page
                                                  Setup...");
        fileMenu.add(pageSetupMenuItem);

        JMenuItem printMenuItem = new JMenuItem("Print...");
        fileMenu.add(printMenuItem);

        fileMenu.addSeparator();
        JMenuItem exitMenuItem = new JMenuItem("Exit");
        fileMenu.add(exitMenuItem);

        //Adding MenuItem to the 'Edit' Menu
        JMenuItem undoMenuItem = new JMenuItem("Undo");
        editMenu.add(undoMenuItem);

        editMenu.addSeparator();

        JMenuItem cutMenuItem = new JMenuItem("Cut");
        editMenu.add(cutMenuItem);

        JMenuItem copyMenuItem = new JMenuItem("Copy");
        editMenu.add(copyMenuItem);

        JMenuItem pasteMenuItem = new JMenuItem("Paste");
```

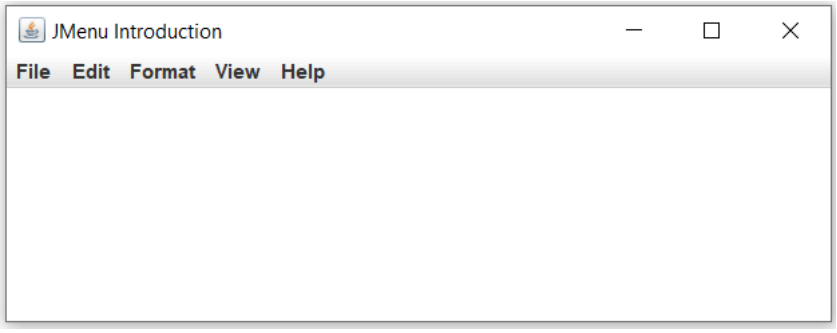
```
        editMenu.add(pasteMenuItem);

        JMenuItem deleteMenuItem= new JMenuItem("Delete");
        editMenu.add(deleteMenuItem);

        //Adding MenuItem to the 'Format' Menu
        JCheckBox wordWrapMenuItemCheckBox = new
                                    JCheckBox("Word Wrap");
        formatMenu.add(wordWrapMenuItemCheckBox);

        JMenuItem fontMenuItem = new JMenuItem("Font..");
        formatMenu.add(fontMenuItem);

        //Adding MenuItem to the 'View' Menu
        JMenu zoomMenu = new JMenu("Zoom");

    //******* Adding MenuItem to the zoomMenu. *************
        JMenuItem zoomInMenuItem = new JMenuItem("Zoom In");
        zoomMenu.add(zoomInMenuItem);

        JMenuItem zoomOutMenuItem = new JMenuItem("Zoom
                                                        Out");
        zoomMenu.add(zoomOutMenuItem);

        JMenuItem defaultZoomMenuItem = new
                          JMenuItem("Restore Default Zoom");
        zoomMenu.add(defaultZoomMenuItem);

//*********************************************************

//Adding MenuItem to the 'View' Menu of type checkbox.
        JCheckBox statusBarCheckBox = new JCheckBox("Status
Bar");
        viewMenu.add(statusBarCheckBox);
        statusBarCheckBox.setSelected(true);
        statusBarCheckBox.setEnabled(false);

        //Adding MenuItem to the 'Help' Menu
        JMenuItem viewHelpMenuItem = new JMenuItem("View
Help");
        helpMenu.add(viewHelpMenuItem);

        JMenuItem sendFeedbackMenuItem = new JMenuItem("Send
                                                Feedback");
        helpMenu.add(sendFeedbackMenuItem);

        helpMenu.addSeparator();

        JMenuItem aboutNotepadMenuItem = new JMenuItem
                                        ("About Notepad");
        helpMenu.add(aboutNotepadMenuItem);
```

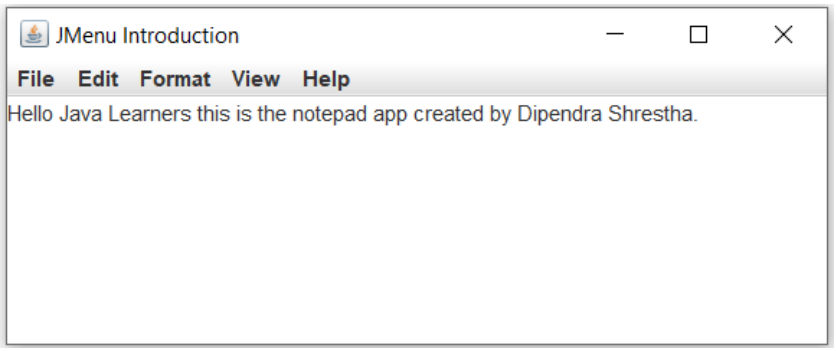
```
//************ Adding ActionListener for 'File' Menu.*****


        // Adding Action Listener for 'New' MenuItem.
        newMenuItem.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                textArea.setText("");
            }
        });

        //Add ActionListener For 'Save' MenuItem.
        saveMenuItem.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                String text = textArea.getText();
                try
                {
                    saveFile(text);
                }
                catch(Exception execption)
                {
                    System.out.println(execption);
                }
            }
        });

        //Adding ActionListener For 'Open' MenuItem.
        openMenuItem.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent
                                                actionEvent)
            {
                try
                {
                    openFile();
                }
                catch (Exception exception)
                {
                    System.out.println(exception);
                }
            }
        });

        //Adding Action Listner for 'Exit' MenuItem.
        exitMenuItem.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                System.exit(0);
```

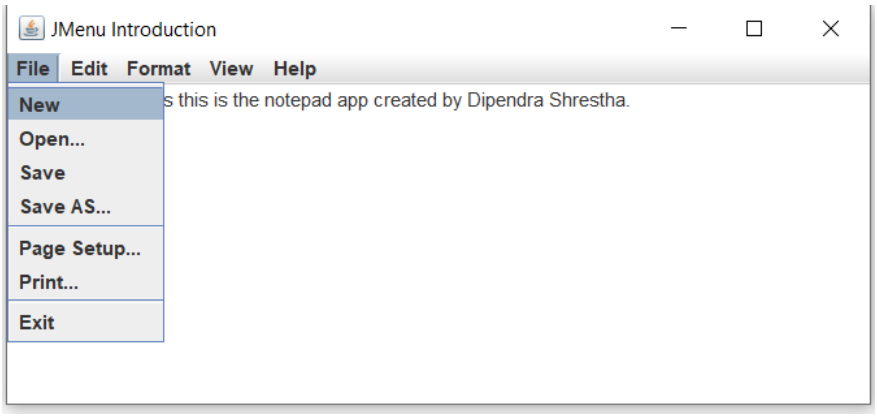
```
            }
        });

        setDefaultCloseOperation(EXIT_ON_CLOSE);

    }

//***************** For 'Save'  menuItem functionality*****


    private void saveFile(String text) throws IOException
    {
        String userDir = System.getProperty("user.home");
        JFileChooser chooser = new JFileChooser
                                        (userDir+"/Desktop");

        chooser.setFileFilter(new FileNameExtensionFilter
                            ("Text Files (*.txt)", "txt"));
        chooser.setSelectedFile(new File(".txt"));
        int result = chooser.showSaveDialog(this);

        if(result == JFileChooser.APPROVE_OPTION)
        {
            File file = chooser.getSelectedFile();
            PrintWriter out = null;
            try
            {
                out = new PrintWriter(file);
                out.print(text);
            }
            finally
            {
                out.close();
            }
        }
        else
            return;
    }

//****************For 'Open' menuItem functionality*********

    private void openFile() throws Exception
    {
        String userDir = System.getProperty("user.home");
        JFileChooser fileChooser = new JFileChooser
                                        (userDir+"/Desktop");

       fileChooser.setFileFilter(new FileNameExtensionFilter
                            ("Text Files(*.txt)","txt"));

        int result = fileChooser.showOpenDialog(this);

        if(result == JFileChooser.APPROVE_OPTION)
        {
            File selectedFile=fileChooser.getSelectedFile();
            BufferedReader in = null;
```

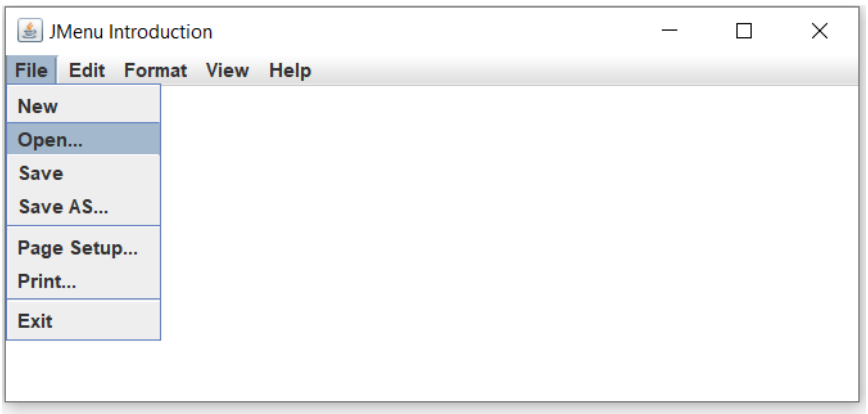
```
            String fileName = selectedFile.getName();
            setTitle(fileName);

            try
            {
                in = new BufferedReader(new FileReader
                                        (selectedFile));
                StringBuilder sb = new StringBuilder();

                String line;
                while (true)
                {
                    line = in.readLine();
                    sb.append(line + "\n");
                    if (line==null)
                        break;
                    textArea.setText(sb.toString());
                }
            }
            finally
            {
                if (in!=null)in.close();
            }
        }
    }
}
```

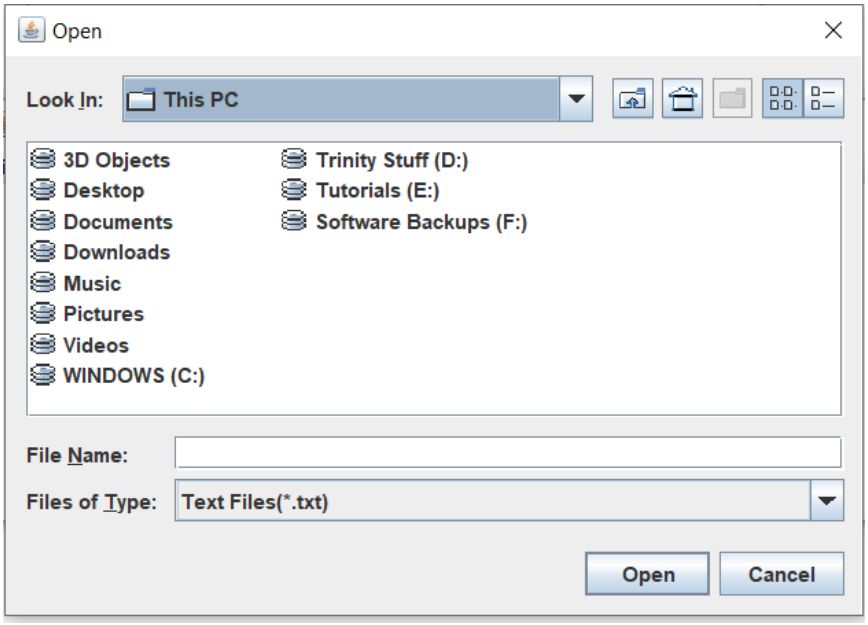
Output

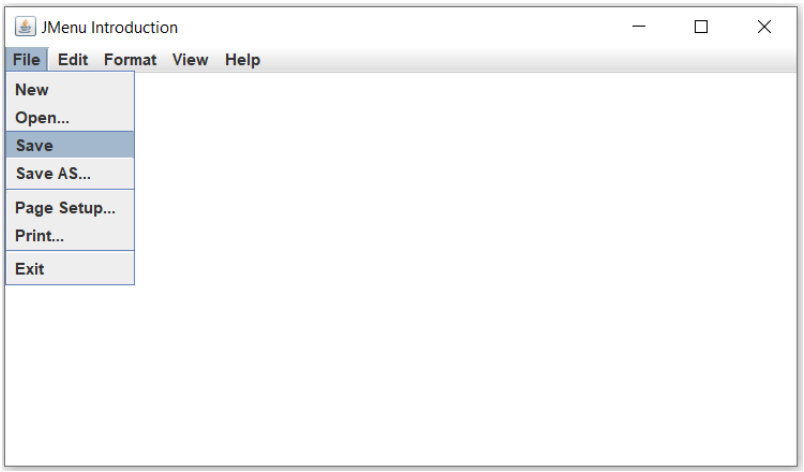


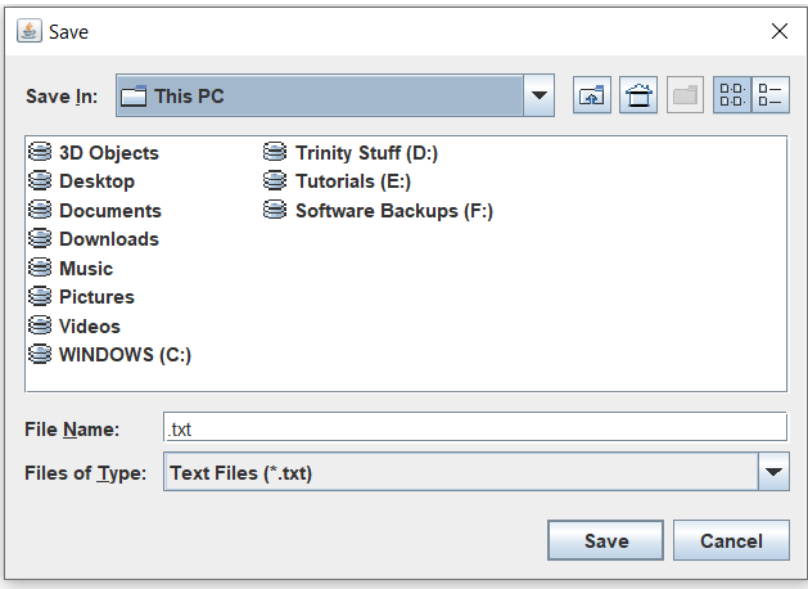


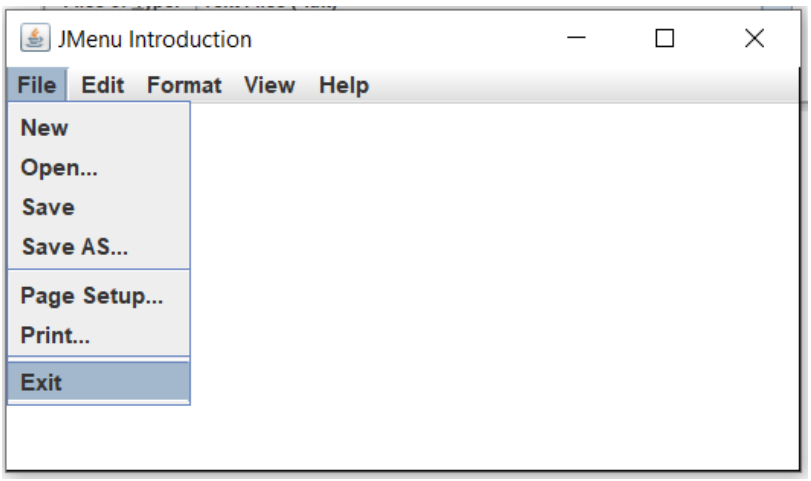

JMenu Introduction
File
Edit
Format
View
Help
New
Open...
Save
Save AS...
Page Setup...
Print...
Exit
s this is the notepad app created by Dipendra Shrestha.


JMenu Introduction
File
Edit
Format
View
Help
New
Open...
Save
Save AS...
Page Setup...
Print...
Exit


Open
Look In:
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
WINDOWS (C:)
Trinity Stuff (D:)
Tutorials (E:)
Software Backups (F:)
File Name:
Files of Type:
Text Files(*.txt)
Open
Cancel


JMenu Introduction
File
Edit
Format
View
Help
New
Open...
Save
Save AS...
Page Setup...
Print...
Exit


Save
Save In:
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
WINDOWS (C:)
Trinity Stuff (D:)
Tutorials (E:)
Software Backups (F:)
File Name:
.txt
Files of Type:
Text Files (*.txt)
Save
Cancel


JMenu Introduction
File
Edit
Format
View
Help
New
Open...
Save
Save AS...
Page Setup...
Print...
Exit

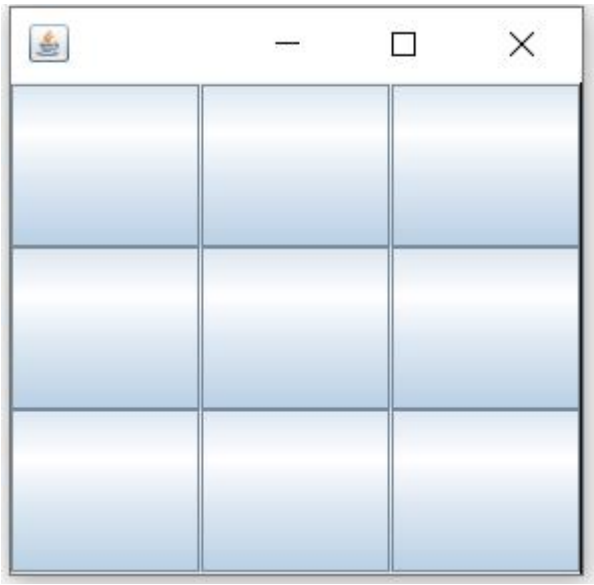
10) Create the UI for tic-tac-toe app using `JButton` array and `GridLayout`.

⇨

Program

```
package Q10_TicTacToeApp;

import javax.swing.*;
import java.awt.*;

public class TicTacToe extends JFrame
{
    public static void main(String[] args)
    {
        TicTacToe frame = new TicTacToe();
        frame.setBounds(500,100,500,500);
        frame.setVisible(true);
    }
    public TicTacToe()
    {
        JButton[] JButtonArray = new JButton[9];
        setLayout(new GridLayout(3,3));

        for (int i=0; i<9; i++)
        {
            JButtonArray[i] = new JButton();
            add(JButtonArray[i]);
        }
        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

Output



11) Demonstrate the use of **Open** and **Save** dialogs for opening and saving files.

⇨

Program

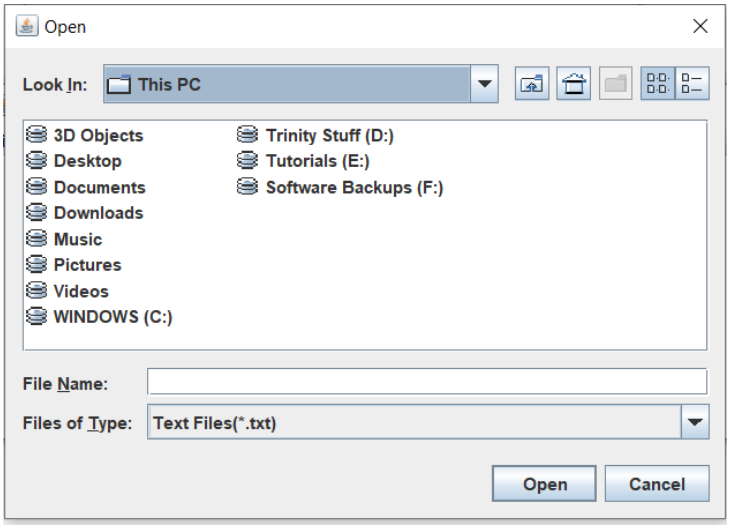
```
package Q11_OpenAndSaveDialog;

import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;

public class OpenAndSaveDemo extends JFrame
{
    public static void main(String[] args)
    {
        OpenAndSaveDemo frame = new OpenAndSaveDemo();
        frame.setBounds(500,100,1000,500);
        frame.setVisible(true);
    }
    JTextArea textArea;
    public OpenAndSaveDemo()
    {

        textArea = new JTextArea();
        add(textArea);
        JMenuBar menuBar = new JMenuBar();
        setJMenuBar(menuBar);

        JMenu fileMenu = new JMenu("File");
        menuBar.add(fileMenu);

        JMenuItem openMenuItem = new JMenuItem("Open");
        JMenuItem saveMenuItem = new JMenuItem("save");
        fileMenu.add(openMenuItem);
        fileMenu.add(saveMenuItem);

        openMenuItem.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent
                                                actionEvent)
            {
                try                 {
                    openFile();
                }
                catch (Exception exception)
                {
                    System.out.println(exception);
                }
            }
        });
```

```
        saveMenuItem.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                String text = textArea.getText();
                try                 {
                    saveFile(text);
                }
                catch(Exception execption)
                {
                    System.out.println(execption);
                }
            }
        });
        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }


    private void openFile() throws Exception
   {
        String userDir = System.getProperty("user.home");
        JFileChooser fileChooser = new JFileChooser
                                    (userDir+"/Desktop");

      fileChooser.setFileFilter(new FileNameExtensionFilter
                              ("Text Files(*.txt)","txt"));

        int result = fileChooser.showOpenDialog(this);

        if(result == JFileChooser.APPROVE_OPTION)
        {
            File selectedFile =fileChooser.
                                        getSelectedFile();

            BufferedReader in = null;
            String fileName = selectedFile.getName();
            setTitle(fileName);

            try{
                in = new BufferedReader(new FileReader
                                        (selectedFile));
                StringBuilder sb = new StringBuilder();

                String line;
                while (true)
                {
                    line = in.readLine();
                    sb.append(line + "\n");
                    if (line==null)
                        break;
                    textArea.setText(sb.toString());
                }
            }
```

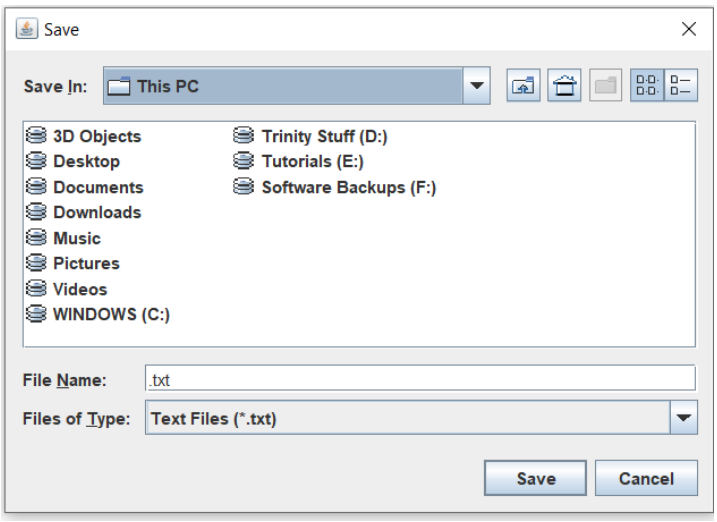
```
            finally
            {
                if (in!=null) in.close();
            }

        }
    }

    private void saveFile(String text) throws IOException
    {
        String userDir = System.getProperty("user.home");
        JFileChooser chooser = new JFileChooser
                                        (userDir+"/Desktop");

        chooser.setFileFilter(new FileNameExtensionFilter
                            ("Text Files (*.txt)", "txt"));

        chooser.setSelectedFile(new File(".txt"));

        int result = chooser.showSaveDialog(this);

        if(result == JFileChooser.APPROVE_OPTION)
        {
            File file = chooser.getSelectedFile();
            PrintWriter out = null;
            try
            {
                out = new PrintWriter(file);
                out.print(text);
            }
            finally
            {
                out.close();
            }
        }
        else
            return;
    }
}
```

Output


File
Open
save


Open
Look In:
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
WINDOWS (C:)
Trinity Stuff (D:)
Tutorials (E:)
Software Backups (F:)
File Name:
Files of Type:
Text Files(*.txt)
Open
Cancel


File
Open
save


Save
Save In:
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
WINDOWS (C:)
Trinity Stuff (D:)
Tutorials (E:)
Software Backups (F:)
File Name:
.txt
Files of Type:
Text Files (*.txt)
Save
Cancel

12) Create a simple app with menus. Include a menu item inside the Help menu to show a custom dialog named AboutDialog. The dialog must contain your App name, version and copyright information, along with a working close button (JButton).

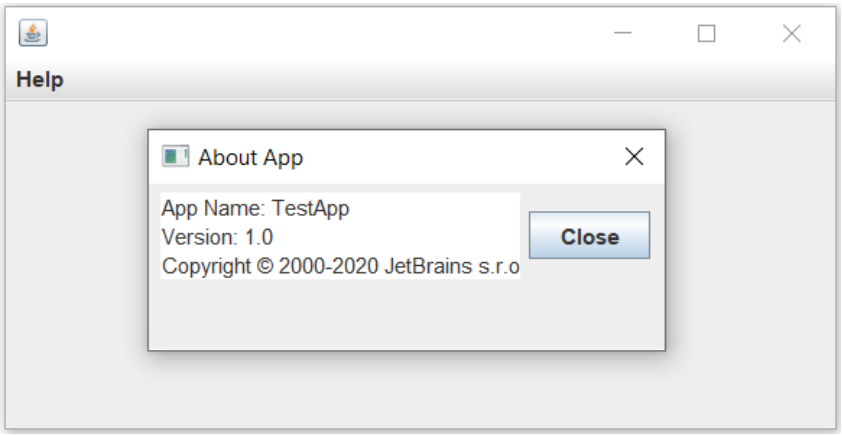
⇨

Program

```
package Q12_SimpleAppWithMenus;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class DialogDemo extends JFrame
{
    public static void main(String[] args)
    {
        DialogDemo frame = new DialogDemo();
        frame.setBounds(500,100,1000,500);
        frame.setVisible(true);
    }
    JDialog dialogBox;
    public DialogDemo()
    {
        JMenuBar menuBar = new JMenuBar();
        setJMenuBar(menuBar);

        JMenu helpMenu = new JMenu("Help");
        menuBar.add(helpMenu);

        JMenuItem aboutDialog = new JMenuItem("About");
        helpMenu.add(aboutDialog);

        JTextArea textArea = new JTextArea();
        textArea.setText("App Name: TestApp"+
                        "\n"+"Version: 1.0"+ "\n"+
                "Copyright © 2000-2020 JetBrains s.r.o");

        textArea.setEditable(false);

        JButton closeButton = new JButton("Close");

        aboutDialog.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent
                                        actionEvent)
            {
                dialogBox = new JDialog
                            (DialogDemo.this,"About App");
```

```
                dialogBox.setLayout(new FlowLayout());
                dialogBox.setBounds(750,250,300,130);
                dialogBox.setVisible(true);

                dialogBox.add(textArea);
                dialogBox.add(closeButton);
                dialogBox.setResizable(false);
            }
        });

        closeButton.addActionListener(new ActionListener()
        {
            @Override
           public void actionPerformed(ActionEvent
                                                actionEvent)
            {
                dialogBox.dispose();
            }
        });

        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

Output



13) Create a form using JFrame to collect the records of students in Trinity. Each record should contain the following information:
    a) First Name (JTextField)
    b) Last Name (JTextField)
    c) Age (JTextField)
    d) Gender (JRadioButton)
    e) Faculty (JComboBox/JList)
    f) Semester (JComboBox/JList)
    g) Remarks (JTextArea)

    Add both menus and toolbars to save the form to a file (display a save dialog). Also add menu/toolbar items to reset the form as well as exit the program. Remember to close the file on exit command.

⇨

Program

```
package Q13_StudentsRecodrs;

import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;

public class StudentsRecord extends JFrame
{
    public static void main(String[] args)
    {
        StudentsRecord frame = new StudentsRecord();
        frame.setVisible(true);
        frame.setBounds(500,100,500,800);
        frame.setTitle("Student Records");

    }

    JTextField firstNameField,lastNameField,ageField;
    JRadioButton maleRadioButton,femaleRadioButton;
    JComboBox facultyComboBox,semesterComboBox;
    JTextArea remarksTextArea;
    ButtonGroup group;
    public StudentsRecord()
    {
        setLayout(new GridLayout(18,0));


//****************** For MenuBar, Menu and MenuItem **************
        JMenuBar menuBar = new JMenuBar();
        menuBar.setBackground(Color.green );
        setJMenuBar(menuBar);

        JMenu fileMenu = new JMenu("File");
        menuBar.add(fileMenu);
```

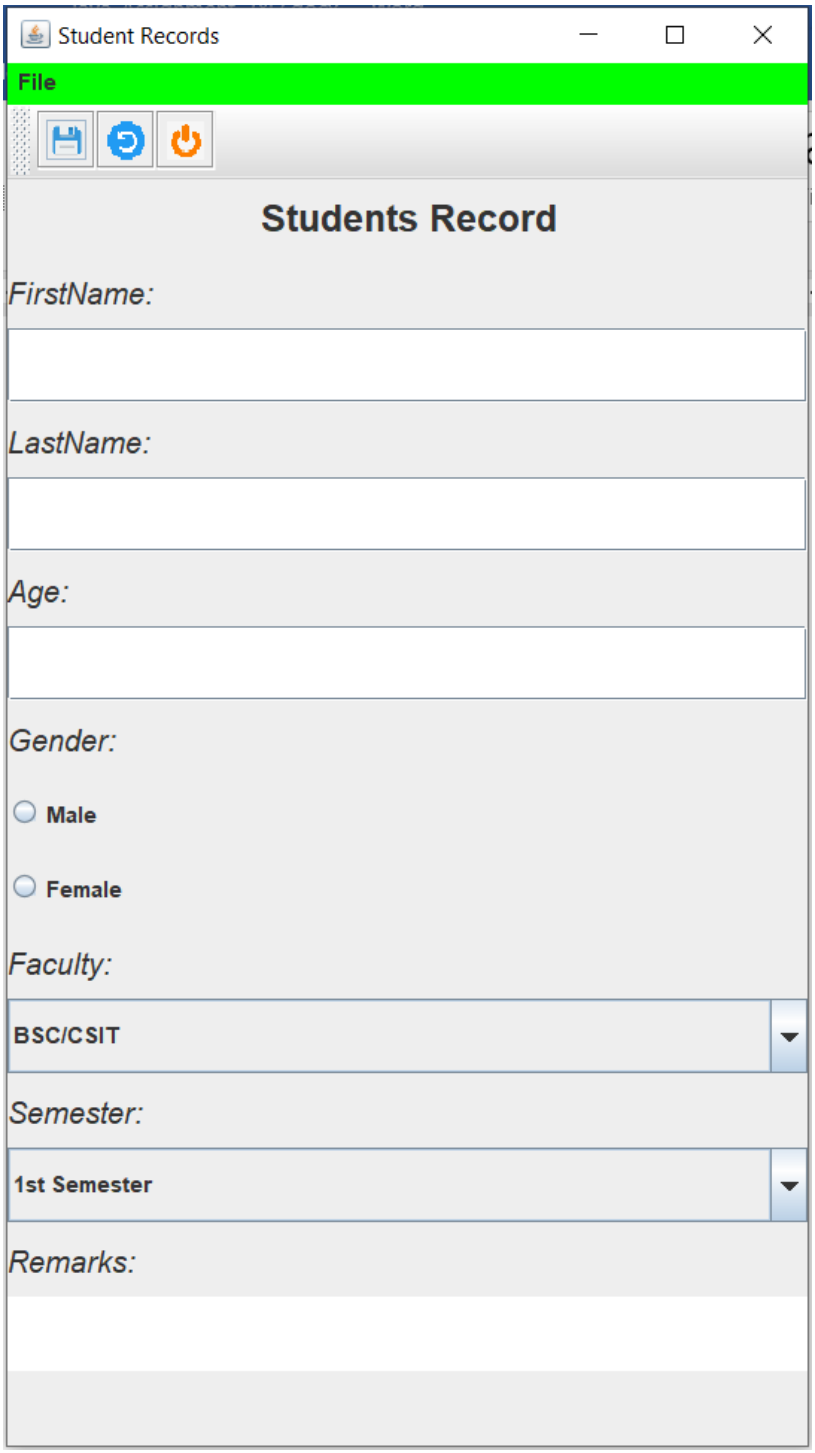
```
        JMenuItem saveMenuItem = new JMenuItem("Save");
        fileMenu.add(saveMenuItem);

        JMenuItem resetMenuItem = new JMenuItem("Reset");
        fileMenu.add(resetMenuItem);

        fileMenu.addSeparator();
        JMenuItem exitMenuItem = new JMenuItem("Exit");
        fileMenu.add(exitMenuItem);


//********************** For JToolBar *************************

        ImageIcon resetIcon = new ImageIcon("resetIcon.png");
        ImageIcon saveIcon = new ImageIcon("saveIcon.png");
        ImageIcon exitIcon = new ImageIcon("exitIcon.png");

        JToolBar toolBar = new JToolBar();
        add(toolBar);

        JButton saveButton = new JButton(saveIcon);
        toolBar.add(saveButton);
        JButton resetButton = new JButton(resetIcon);
        toolBar.add(resetButton);
        JButton exitButton = new JButton(exitIcon);
        toolBar.add(exitButton);


//********** For JTextField,CheckBox and ComboBox ****************

        JLabel title = new JLabel("Students Record");
        Font font = new Font("TimesRoman",Font.BOLD,20);
        title.setFont(font);
        title.setHorizontalAlignment(title.CENTER);
        add(title);

        Font font1 = new Font("TimesRoman",Font.ITALIC,16);
        JLabel firstNameLabel = new JLabel("FirstName:");
        firstNameLabel.setFont(font1);
        firstNameField = new JTextField(10);

        JLabel lastNameLabel = new JLabel("LastName:");
        lastNameLabel.setFont(font1);
        lastNameField = new JTextField(10);

        JLabel ageLabel = new JLabel("Age:");
        ageLabel.setFont(font1);
        ageField = new JTextField(10);

        JLabel genderLabel = new JLabel("Gender:");
        genderLabel.setFont(font1);
        maleRadioButton = new JRadioButton("Male");
        femaleRadioButton = new JRadioButton("Female");

        JLabel facultyLabel = new JLabel("Faculty:");
        facultyLabel.setFont(font1);
        String[] facultyList =
                            {"BSC/CSIT","BBM","BIT","Engineering"};
        facultyComboBox = new JComboBox(facultyList);
```

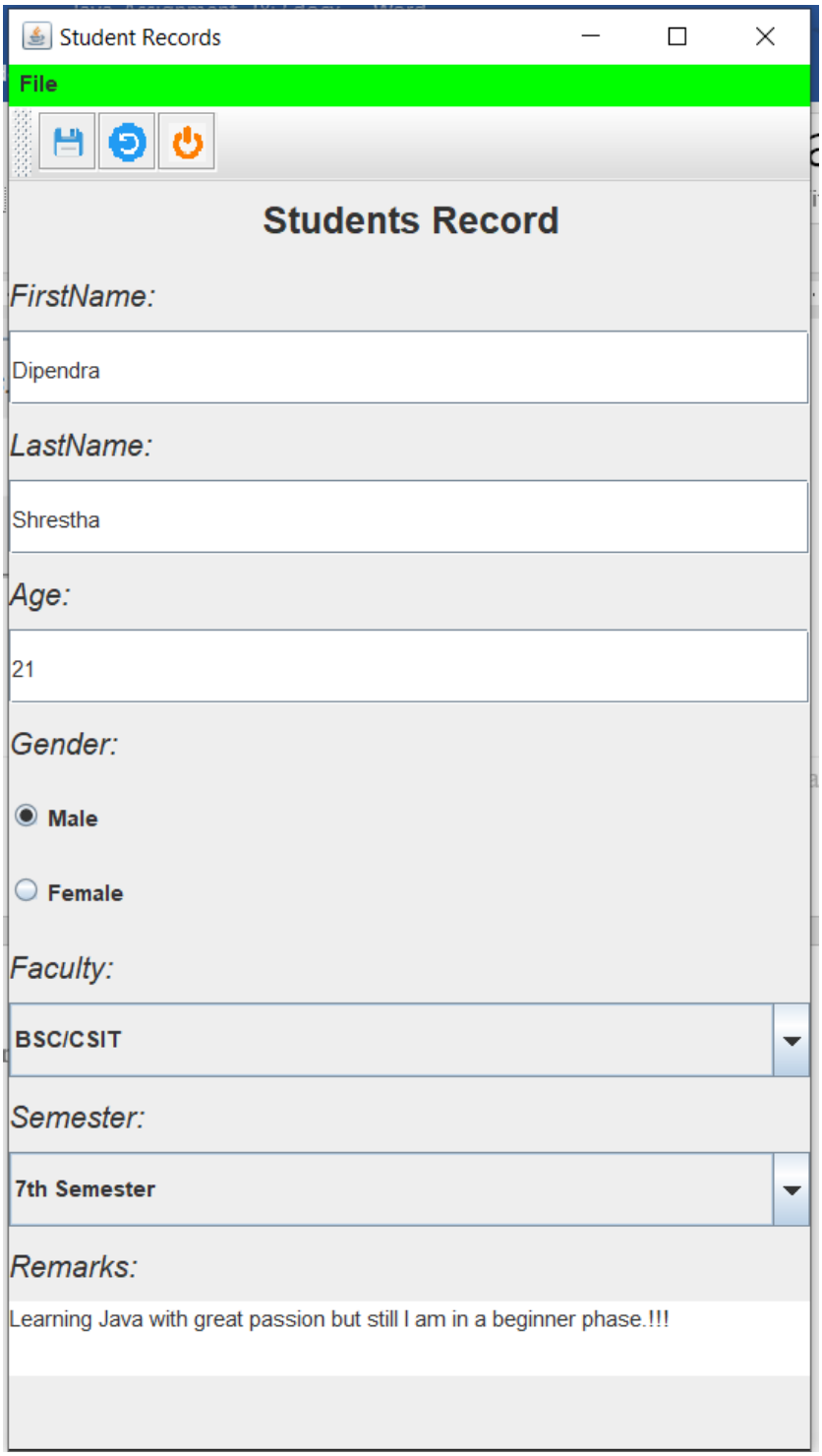
```
        JLabel semesterLabel = new JLabel("Semester:");
        semesterLabel.setFont(font1);
        String[] semesterList ={"1st Semester","2nd Semester",
                                "3rd Semester","4th Semester",
                                "5th Semester","6th Semester",
                                "7th Semester","8th Semester"};

        semesterComboBox = new JComboBox(semesterList);

        JLabel remarksLabel = new JLabel("Remarks:");
        remarksLabel.setFont(font1);
        remarksTextArea = new JTextArea();


//*** Adding Label, Fields,RadioButtons, ComboBox and TextArea****

        add(firstNameLabel);
        add(firstNameField);

        add(lastNameLabel);
        add(lastNameField);

        add(ageLabel);
        add(ageField);
        add(genderLabel);
        group = new ButtonGroup();

        add(maleRadioButton);
        add(femaleRadioButton);
        group.add(maleRadioButton);
        group.add(femaleRadioButton);

        add(facultyLabel);
        add(facultyComboBox);

        add(semesterLabel);
        add(semesterComboBox);

        add(remarksLabel);
        add(remarksTextArea);

//********** ActionListener For JButton on ToolBar ***************

        saveButton.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent actionEvent)
            {
                String[] text = getFieldValue();
                try
                {
                    saveFormData(text);
                }
                catch (Exception e)
                {
                    System.out.println(e.getMessage());
                }
            }
        });
```

```
        resetButton.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent actionEvent)
            {
                resetMethods();

            }
        });

        exitButton.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent actionEvent)
            {
                System.exit(0);
            }
        });

//*********** Action Listener For JMenuItem on Menu **************

        saveMenuItem.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent actionEvent)
            {
                String[] text = getFieldValue();
                try
                {
                    saveFormData(text);
                }
                catch (Exception e)
                {
                    System.out.println(e.getMessage());
                }
            }
        });


        resetMenuItem.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent actionEvent)
            {
                resetMethods();
            }
        });
        exitMenuItem.addActionListener(new ActionListener()
        {
            @Override
            public void actionPerformed(ActionEvent actionEvent)
            {
                System.exit(0);
            }
        });

        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
```

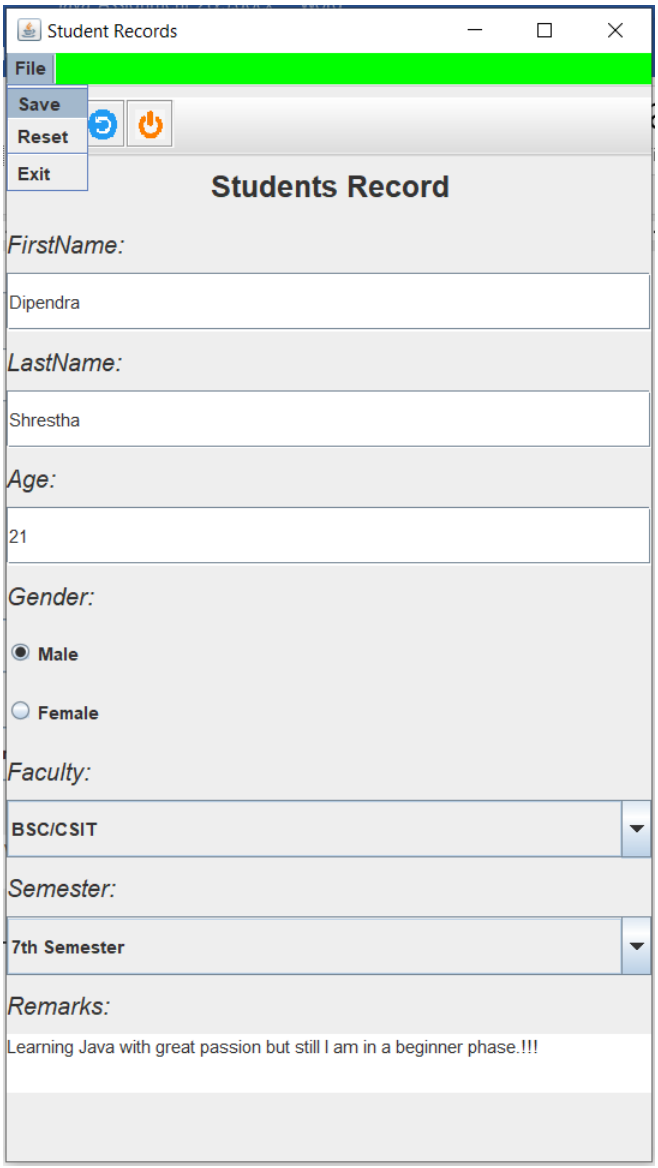
```
//*********** Methods for Reset and Save *********************
    private void resetMethods()
    {
        firstNameField.setText("");
        lastNameField.setText("");
        ageField.setText("");
        group.clearSelection();
        facultyComboBox.setSelectedIndex(0);
        semesterComboBox.setSelectedIndex(0);
        remarksTextArea.setText("");
    }
    private String[] getFieldValue()
    {
        maleRadioButton.setActionCommand("Male");
        femaleRadioButton.setActionCommand("Female");

        String[] text = {
                firstNameField.getText(),
                lastNameField.getText(),
                ageField.getText(),
                group.getSelection().getActionCommand(),
                (String)facultyComboBox.getSelectedItem(),
                (String)semesterComboBox.getSelectedItem(),
                remarksTextArea.getText()
        };
        return text;
    }
    private void saveFormData (String[] text) throws IOException
    {
        String userDir = System.getProperty("user.home");
        JFileChooser chooser = new JFileChooser
                                            (userDir+"/Desktop");

        chooser.setFileFilter(new FileNameExtensionFilter("Text
                                        Files (*.txt)", "txt"));
        chooser.setSelectedFile(new File(".txt"));
        int result = chooser.showSaveDialog(this);

        if(result == JFileChooser.APPROVE_OPTION)
        {
            File file = chooser.getSelectedFile();
            PrintWriter out = null;
            try
            {
                out = new PrintWriter(file);
                for (int i=0; i<text.length; i++)
                {
                   out.print(text[i]+"\n");
                }
            }
            finally
            {
                out.close();
            }
        }
        else
            return;
    }

}
```

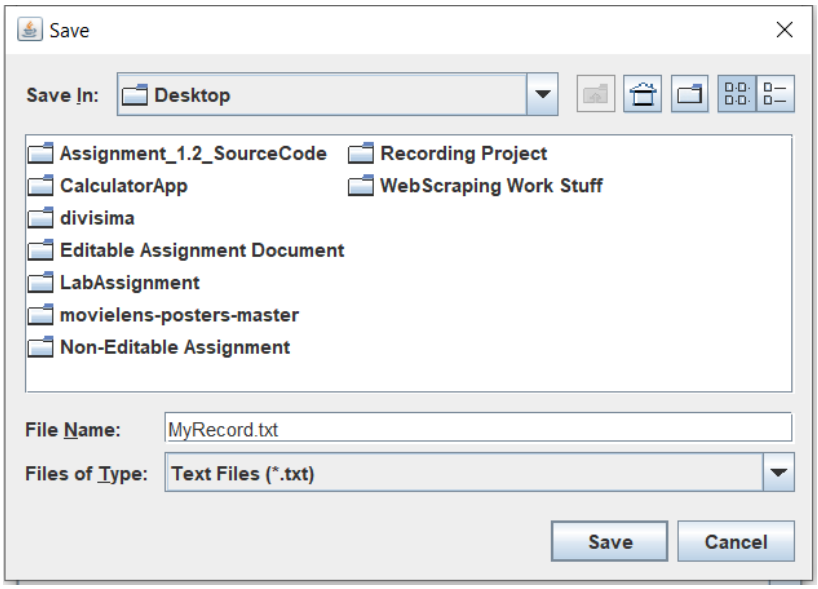
Output


Student Records
File
Students Record
FirstName:
LastName:
Age:
Gender:
Male
Female
Faculty:
BSC/CSIT
Semester:
1st Semester
Remarks:

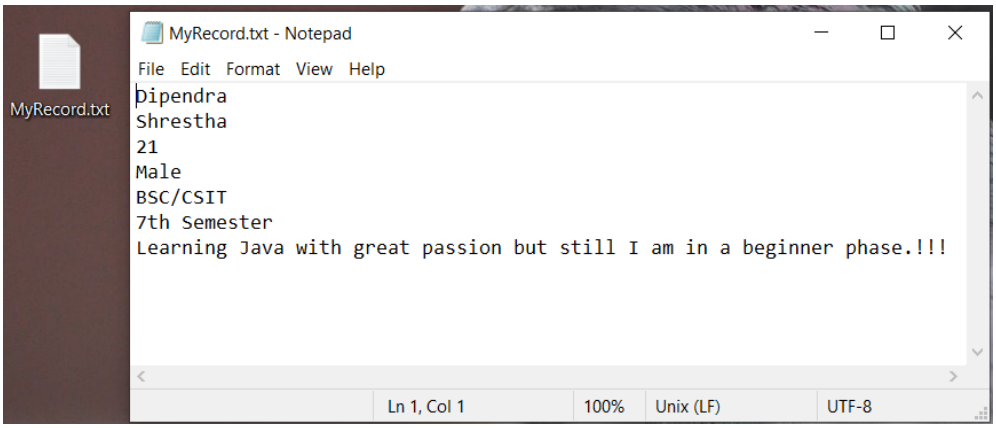

Student Records
File
Students Record
FirstName:
Dipendra
LastName:
Shrestha
Age:
21
Gender:
Male
Female
Faculty:
BSC/CSIT
Semester:
7th Semester
Remarks:
Learning Java with great passion but still I am in a beginner phase.!!!

Either you can save from shortcut icon as well. It works!!





As we can see that record has been successfully saved to desktop.



If we press 'Reset' menu item then our form field will be reset.