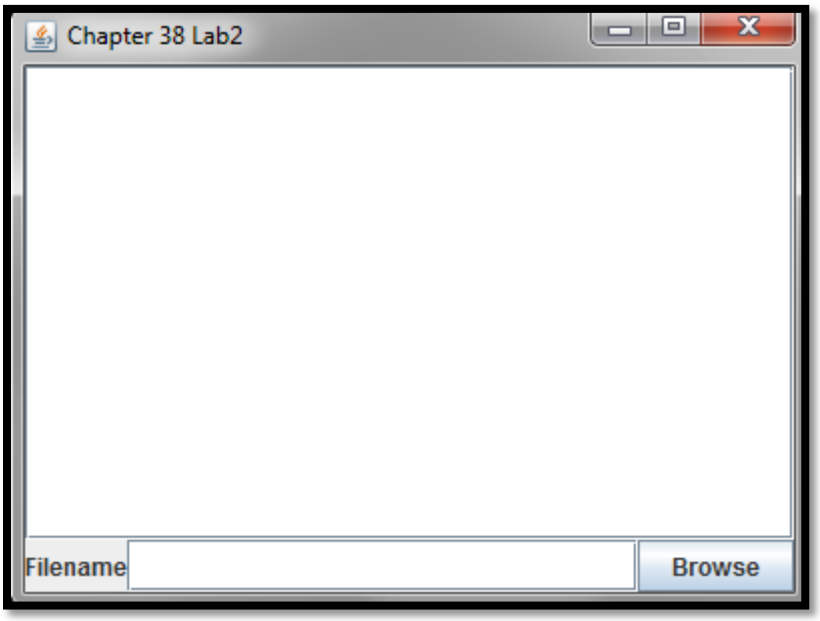
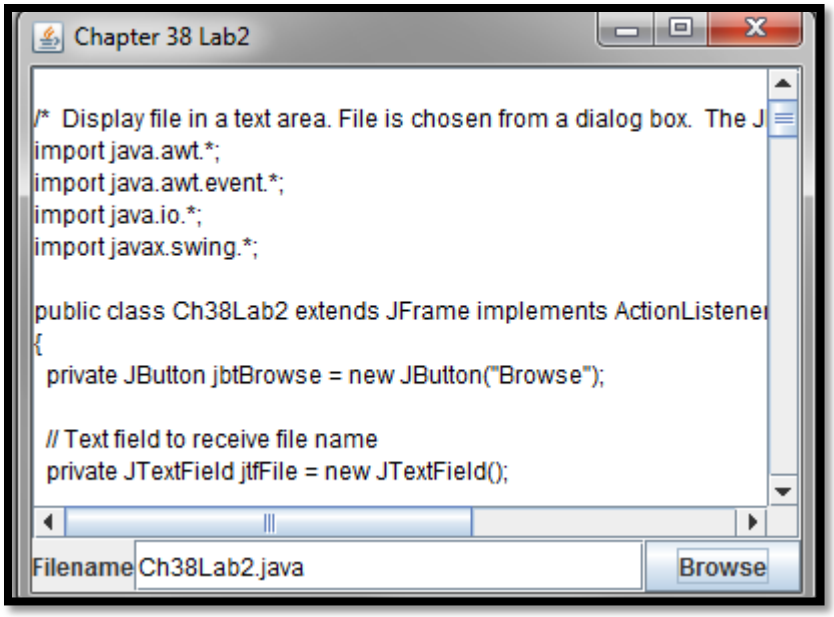
CIT 249: Java II
Chapter 38 Lab2

In this lab we will create a program that enables the user to select a file from a file open dialog box. A file open dialog box is displayed when the Browse button is clicked. The file is displayed in the text area, and the file name is displayed in the text field when the OK button is clicked in the file open dialog box. You can also enter the file name in the text field and press the Enter key to display the file in the text area. The following are screenshots:





Here we browse to a file and it is displayed within the window.

Let's get started.

1. Open a new document window and save the file as Ch38Lab2.java. Our purpose is:

   /* Display file in a text area. File is chosen from a dialog box. The JFileChooser class will be demonstrated. */

2. Type the import statements that will import all classes in the java.awt, java.awt.event, java.io, and javax.swing packages.
3. Type the class header and opening brace having the class extend the JFrame class and implement ActionListener interface.
4. Next we'll declare several components. Type:

   ```
   //construct several objects
   private JButton jbtBrowse = new JButton("Browse");

    // Text field to receive file name
    private JTextField jtfFile = new JTextField();

    // Text area to display file
    private JTextArea jtaFileContent = new JTextArea();
   ```

```
// Create jFileChooser
private JFileChooser jFileChooser = new JFileChooser();
```

- The JFileChooser class allows you to browse to a file and select it.
- The JTextArea will display the contents of the file.
- The JButton will be used to do the actual browsing.

5. Next is our constructor method. Type:

```
public Ch38Lab2()
{
  // Create a Panel to hold a label, a text field, and a button
  JPanel p = new JPanel();
  p.setLayout(new BorderLayout());
  p.add(new JLabel("Filename"), BorderLayout.WEST);
  p.add(jtfFile, BorderLayout.CENTER);
  jtfFile.setBackground(Color.white);
  jtfFile.setForeground(Color.black);
  p.add(jbtBrowse, BorderLayout.EAST);
```

- We create a new panel and set its layout to BorderLayout.
- We add a new label to the WEST section of the panel.
- We add the text field to the panel's CENTER.
- We set the text field's background color to white and foreground color to black.
- We add the button to the EAST section of the panel.

```
  // Create a scrollabel text area
  JScrollPane jsp = new JScrollPane(jtaFileContent);
  // Set default directory to the current directory

  jFileChooser.setCurrentDirectory(new File("."));

  // Use BorderLayout for the frame
  setLayout(new BorderLayout());

  add(jsp, BorderLayout.CENTER);
  add(p, BorderLayout.SOUTH);
  jtaFileContent.setBackground(Color.white);

 jtaFileContent.setForeground(Color.black);

  // Register listener
  jbtBrowse.addActionListener(this);
  jtfFile.addActionListener(this);
}
```

- We construct a new JScrollPane object and add the text area to it.
- We invoke the JFileChooser's setCurrentDirectory() method passing to it a new file
- We set the frame's layout to BorderLayout.
- We add the scroll pane to the frame's center.
- We add our panel to the frame's SOUTH.
- We set the background color of the text area to while with a black foreground.
- We add a listener to the text field and the button.

6. Next is our actionPerformed() method. Type:

```
public void actionPerformed(ActionEvent e)
{
    //if the jbtBrowse button is selected, invoke the browse() method, else invoke
    //showFile() method passing to it the value in the text field
   if (e.getSource() == jbtBrowse)
   {
     browse();
   }
   else if (e.getSource() == jtfFile)
   {
     showFile(new File(jtfFile.getText().trim()));
   }
 }
```

- If the button is pressed we invoke the browse() method.
- If the text field is selected and the Enter key is pressed the showFile() method is invoked and a new File object is created. The file is what has been entered in the text field with all white space trimmed away.

7. The next method is the browse() method. Type:

```
private void browse()
{
    //if the open file is selected get the selected file
   if (jFileChooser.showOpenDialog(this) == JFileChooser.APPROVE_OPTION)
   {
     showFile(jFileChooser.getSelectedFile());
   }
  }
```

- If what is returned by the showOpenDialog() method equals the JFileChooser's APPROVE_OPTION, then the showFile() method is invoked, passing to it the contents of the file.

8. The final method is the showFile() method. Type:

```
private void showFile(File file)
{

    BufferedReader infile = null;  //declare buffered stream

   //get file name from the text field
   String inLine;

   jtfFile.setText(file.getName());
```

- A new BufferedReader object is created and set to a null value.
- A String variable is declared that will get the file name from the text field.
- The text field has its text set to the name of the file.

```
try
{

  //create a buffered stream
  infile = new BufferedReader(new FileReader(file));

  //read a line
  inLine = infile.readLine();

  boolean firstLine = true;

  //append the line to the text area
  while (inLine != null)
  {

    if (firstLine)
    {
      firstLine = false;
      jtaFileContent.append(inLine);
    }
    else
    {
      jtaFileContent.append("\n" + inLine);
    }

    inLine = infile.readLine();
  }
}
```

- A buffered stream is created
- A line is read
- While the String is not null each line is added to the text area.

```
//Throw an exception if the file is not found
     catch (FileNotFoundException ex)

    {

      System.out.println("File not found: " + file.getName());

    }

     //if there are problems with input/output catch an exception
    catch (IOException ex)

    {

      System.out.println(ex.getMessage());

    }

    finally

    {

      try

      {

        if (infile != null) infile.close();

      }

      catch (IOException ex)

      {

      }

    }

  }
```

9. Finally the main method. Type:

```
public static void main(String[] args)
{
   Ch38Lab2 frame = new Ch38Lab2();
   frame.setSize(400, 300);
   frame.setTitle("Chapter 38 Lab2");
   frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   frame.setLocationRelativeTo(null); // Center the frame
   frame.setVisible(true);
 }
```

10. Close the class.
11. Compile the program and fix any errors if necessary.
12. Compress all files into a single zip or rar file and submit.