```java
import com.toedter.calendar.JCalendar;

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

public class Main {
   public static void main(String[] args) {

       Frame f = new Frame();

   }
}
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import javax.swing.ImageIcon;

public class Frame {

   private JFrame f;

   public Frame(){
       f = new JFrame();

       f.setSize(750,500);
       f.setLocationRelativeTo(null);
       f.setLayout(new GridBagLayout());
       f.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
       f.getContentPane().setBackground(Color.LIGHT_GRAY);

       GridBagConstraints g = new GridBagConstraints();
       g.fill = GridBagConstraints.BOTH;
       g.insets=new Insets(10, 10, 10, 10);

       JLabel Header = new JLabel("Lesson Planner", SwingConstants.CENTER);
       Header.setFont(new Font("Arial", Font.BOLD, 60));
       Header.setHorizontalAlignment(SwingConstants.CENTER);
       g.gridx =0;
       g.gridy=0;
       g.gridwidth =3;
       g.anchor = GridBagConstraints.CENTER;
       f.add(Header, g);


       ImageIcon icon = new ImageIcon("Images/PLannerICON.png");
       Image scaled = icon.getImage().getScaledInstance(200, 200,
Image.SCALE_SMOOTH);
       ImageIcon resized = new ImageIcon(scaled);
       JLabel imageLabel = new JLabel(resized);
       g.gridx = 1;
       g.gridy = 2;
       g.gridwidth = 1;
       g.anchor = GridBagConstraints.CENTER;
       f.add(imageLabel,g);

       JButton schedule = new JButton("Schedule");
       schedule.setPreferredSize(new Dimension(200, 50));
       g.gridy = 3;
       g.gridx = 0;
```

```
        g.gridwidth = 1;
        //g.insets = new Insets(10, 20, 10, 10);
        g.anchor = GridBagConstraints.WEST;
        f.add(schedule, g);

        JButton calendar = new JButton("Calendar");
        calendar.setPreferredSize(new Dimension(150, 50));
        g.gridy = 3;
        g.gridx = 1;
        g.gridwidth = 1;
        g.anchor = GridBagConstraints.CENTER;
        f.add(calendar, g);

        JButton savedplans = new JButton("Templates");
        savedplans.setPreferredSize(new Dimension(200, 50));
        g.gridy = 3;
        g.gridx = 2;
        g.gridwidth = 1;
        //g.insets = new Insets(10, 10, 10, 20);
        g.anchor = GridBagConstraints.EAST;
        f.add(savedplans, g);




        f.setVisible(true);

        schedule.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if(e.getSource().equals(schedule)){
                    savedplans.setVisible(false);
                    schedule.setVisible(false);
                    Header.setVisible(false);
                    imageLabel.setVisible(false);
                    calendar.setVisible(false);
                    new Schedule(f);


                }
            }
        });

        savedplans.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if(e.getSource().equals(savedplans)){

                    new PlanTemplate();
```

```
                }
            }
        });

        calendar.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if(e.getSource()==calendar){

                    Calender c = new Calender();


                }
            }
        });

    }

}
```

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

public class Schedule{
   public Schedule(JFrame f) {

       JPanel panel = new JPanel(new GridBagLayout());
       GridBagConstraints gbc = new GridBagConstraints();


       gbc.fill = GridBagConstraints.BOTH; //fills the entire cell,
horizontally and vertically
       gbc.weightx = 1.0; //expands the component
       gbc.weighty = 1.0;
       gbc.insets = new Insets(5, 5, 5, 5);  //adds padding

       String[] days = {"Monday", "Tuesday", "Wednesday", "Thursday",
"Friday"};

       for(int col =0;col<5;col++){ //loop through day labels
           gbc.gridx = col;
           gbc.gridy = 0;
           gbc.gridheight = 1;
           JLabel label = new JLabel(days[col], SwingConstants.CENTER);
           panel.add(label, gbc);
       }



       for(int row=0;row<4;row++){ //loop through buttons
           for(int col = 0;col<5; col++){
               gbc.gridx = col;
               gbc.gridy =row +1;
               JButton button=new JButton("Period "+(row+1));
               button.setPreferredSize(new Dimension(130,100));

               final String day=days[col];
               final int lessonnum= row;


               button.addActionListener(new ActionListener() { //each button
needs an action listener
                   @Override
                   public void actionPerformed(ActionEvent e){ //creates a
lessonplan object when clicked
                       LessonPlan l = new LessonPlan(day,lessonnum+1, null);
```

```
                    }
                });


                panel.add(button, gbc);
            }
        }


        f.add(panel);
        f.setVisible(true);
    }

}
```

```java
import com.toedter.calendar.JCalendar;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.beans.PropertyChangeEvent;
import java.beans.PropertyChangeListener;
import java.util.Date;

public class Calender {


public Calender(){

   JFrame calenderframe = new JFrame("Calender");
   calenderframe.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
   calenderframe.setSize(800, 600);

   JCalendar calender = new JCalendar();
   //add a propertychangelistener to getdaychooser, a component of JCalender
   //daychooser records and updates the most recent day clicked
   //the propertychangelistener will react to any changes in daychooser.
   calender.getDayChooser().addPropertyChangeListener("day", new
PropertyChangeListener() {
       @Override
       public void propertyChange(PropertyChangeEvent evt) {
           Date selectedDate = calender.getDate(); //this method retrieves the
most recently selected date
           makeframe(selectedDate);


       }
   });
   calenderframe.add(calender);

   calenderframe.setVisible(true);

}
public void makeframe(Date date){
//dayframe is only 1 column of 4 buttons, each one a different period
   JFrame dayframe = new JFrame(String.valueOf(date));
   dayframe.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
   dayframe.setSize(160, 550);
   dayframe.setLayout(new GridBagLayout());
   GridBagConstraints gbc = new GridBagConstraints();

   gbc.gridx=0;
   for(int row=0;row<4;row++){ //looped button creation minimises code
```

```
        gbc.gridy=row;
        JButton button=new JButton("Period "+(row+1));
        button.setPreferredSize(new Dimension(130,100));

        int r = row;
        button.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                LessonPlan l = new LessonPlan("", r +1,date);
            }
        });
        dayframe.add(button, gbc);
    }


    dayframe.setVisible(true);

}

}
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.text.SimpleDateFormat;
import java.util.Date;

public class LessonPlan {

   JFrame lessonFrame;
   JTextField subjectfield;
   JTextField goalfield;
   JTextField durationfield;
   JTextArea materialsarea;
   JTextArea planarea;
   JTextArea notesarea;
   JButton clear;
   JButton save;
   JButton load;

   private int lessonnum;
   private String day;
   private static JFrame currentLessonFrame;


   public LessonPlan(String d, int l, Date date){
       this.day=d;
       this.lessonnum=l;

       if (currentLessonFrame != null){
           currentLessonFrame.dispose();
       }
       createLessonWindow(date);
   }
   public void createLessonWindow(Date date){

       if(date!=null){ //plan pages created from the calendar page need a
specific name
           lessonFrame = new JFrame("Period: "+lessonnum+ " ;"+date);
       }else{
           lessonFrame = new JFrame(day+" - Period: "+lessonnum);
       }

       lessonFrame.setSize(800,600);
       lessonFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
       lessonFrame.setLayout(new GridBagLayout());
       GridBagConstraints gbc = new GridBagConstraints();
       gbc.insets = new Insets(5, 5, 5, 5);
```

```
gbc.fill = GridBagConstraints.HORIZONTAL;
//now adding all the components:
JLabel subjectlabel = new JLabel("Subject:");
gbc.gridx= 0;
gbc.gridy=0;
gbc.gridwidth =1;
lessonFrame.add(subjectlabel, gbc);
subjectfield = new JTextField(12);
subjectfield.setPreferredSize(new Dimension(150,25));
gbc.gridx =1;
gbc.gridy= 0;
gbc.gridwidth=2;
lessonFrame.add(subjectfield, gbc);


JLabel goallabel = new JLabel("Lesson goal:");
gbc.gridx=0;
gbc.gridy=1;
gbc.gridwidth=1;
gbc.weightx=0;
lessonFrame.add(goallabel,gbc);
goalfield = new JTextField(15);
goalfield.setPreferredSize(new Dimension(250, 25));
gbc.gridx = 1;
gbc.gridy=1;
gbc.gridwidth=2;
lessonFrame.add(goalfield, gbc);

JLabel duration =new JLabel("Duration:");
gbc.gridx = 3;
gbc.gridy = 1;
gbc.gridwidth = 1;
lessonFrame.add(duration, gbc);
durationfield=new JTextField(6);
durationfield.setPreferredSize(new Dimension(100,25));
gbc.gridx = 5;
gbc.gridy = 1;
gbc.gridwidth = 1;
gbc.weightx = 0.3;
lessonFrame.add(durationfield, gbc);


JLabel materials = new JLabel("Materials:");
gbc.gridx = 0;
gbc.gridy = 2;
gbc.gridwidth = 2;
gbc.anchor = GridBagConstraints.WEST;
gbc.fill = GridBagConstraints.HORIZONTAL;
```

```
        lessonFrame.add(materials, gbc);
        materialsarea = new JTextArea(5, 13);
        materialsarea.setLineWrap(true);
        materialsarea.setWrapStyleWord(true);
        JScrollPane materialsScroll = new JScrollPane(materialsarea); // Add
ScrollPane for usability
        gbc.gridx = 0;
        gbc.gridy = 3;
        gbc.gridwidth = 2;
        gbc.gridheight = 2;
        gbc.weightx = 0;
        gbc.weighty = 1;
        gbc.fill = GridBagConstraints.BOTH;
        lessonFrame.add(materialsScroll, gbc);

        gbc.weighty = 0;

        JLabel plan = new JLabel("Plan Details:");
        gbc.anchor=GridBagConstraints.CENTER;
        gbc.gridx = 2;
        gbc.gridy = 2;
        gbc.gridwidth=2;
        lessonFrame.add(plan,gbc);
        planarea = new JTextArea(5,13);
        planarea.setLineWrap(true);
        planarea.setWrapStyleWord(true);
        JScrollPane planscroll = new JScrollPane(planarea);
        gbc.gridx = 2;
        gbc.gridy = 3;
        gbc.gridwidth=2;
        lessonFrame.add(planscroll, gbc);

        JLabel notes=new JLabel("Notes:");
        gbc.anchor=GridBagConstraints.CENTER;
        gbc.gridx = 4;
        gbc.gridy = 2;
        gbc.gridwidth = 2;
        lessonFrame.add(notes, gbc);
        notesarea = new JTextArea(5,13);
        notesarea.setLineWrap(true);
        notesarea.setWrapStyleWord(true);
        JScrollPane notescroll = new JScrollPane(notesarea);
        gbc.gridx = 4;
        gbc.gridy = 3;
        gbc.gridwidth=2;
        lessonFrame.add(notescroll, gbc);

        clear = new JButton("Clear");
        save = new JButton("Save");
```

```
        load = new JButton("Load");

        //create seperate panel at the bottom so the buttons dont interfere
with the other compenents

        JPanel buttonPanel = new JPanel(new GridLayout(1, 3, 10, 0)); //(rows,
cols, horizontal gap, vertical gap)


        buttonPanel.add(clear);
        buttonPanel.add(save);
        buttonPanel.add(load);

        //place button panel at bottom, spanning 6 cols
        gbc.gridx = 0;
        gbc.gridy = 5;
        gbc.gridwidth = 6;
        gbc.fill = GridBagConstraints.HORIZONTAL;
        lessonFrame.add(buttonPanel, gbc);

        lessonFrame.setVisible(true);


        final String fileName;
        if(date!=null){ //specific file name for plan saved from Calendar
            SimpleDateFormat simpledate = new SimpleDateFormat("yyyy-MM-dd");
            //removes the time from Date object
            fileName = simpledate.format(date) + "; Period: " + lessonnum +
".txt";
        }else{
            fileName = day+ "; Period: " +lessonnum + ".txt";
        }

        //autoload functionality:
        File file = new File(fileName);
        if (file.exists()) {
            try (BufferedReader reader = new BufferedReader(new
FileReader(file))) {

                subjectfield.setText(reader.readLine());
                goalfield.setText(reader.readLine());
                durationfield.setText(reader.readLine());

                reader.readLine(); //this is the "====" line

                StringBuilder materialscontent = new StringBuilder();
                String line = reader.readLine();
                while (line != null && !line.equals("===")) { //the line must
not be null and must not be "==="
```

```
                    materialscontent.append(line).append("\n");
                    //adds the string as a new line to the stringbuilder object
                    line = reader.readLine();//move to next line
                }
                line =reader.readLine(); //skip the "===" line

                StringBuilder plancontent = new StringBuilder();
                while (line != null && !line.equals("===")) {
                    plancontent.append(line).append("\n");
                    line = reader.readLine();
                }
                line = reader.readLine();

                StringBuilder notescontent = new StringBuilder();
                while (line != null) {
                    notescontent.append(line).append("\n");
                    line = reader.readLine();
                }

                materialsarea.setText(materialscontent.toString()); //now add
the text
                planarea.setText(plancontent.toString());
                notesarea.setText(notescontent.toString());

            } catch (IOException ex) {
                JOptionPane.showMessageDialog(lessonFrame, "Error loading file:
" + ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
        save.addActionListener(new ActionListener()  {
            @Override
            public void actionPerformed(ActionEvent e){
                File file = new File(fileName);

                try(BufferedWriter writer = new BufferedWriter(new
FileWriter(fileName))){
                    writer.write(subjectfield.getText());
                    writer.newLine();
                    writer.write(goalfield.getText());
                    writer.newLine();
                    writer.write(durationfield.getText());
                    writer.newLine();

                    //seperates the large text fields as these could be many
lines
                    writer.write("===");
                    writer.newLine();
                    // write the materials text
                    writer.write(materialsarea.getText());
```

```
                    writer.newLine();

                    writer.write("===");
                    writer.newLine();
                    writer.write(planarea.getText());
                    writer.newLine();

                    writer.write("===");
                    writer.newLine();
                    writer.write(notesarea.getText());

                }catch(IOException ex){
                    JOptionPane.showMessageDialog(lessonFrame, "Error saving
file: "+ex.getMessage(),
                            "Error", JOptionPane.ERROR_MESSAGE);
                    return;
                }
                JOptionPane.showMessageDialog(lessonFrame, "Lesson plan saved",
"Success",
                        JOptionPane.INFORMATION_MESSAGE);
            }
        });
        clear.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {

                int confirm = JOptionPane.showConfirmDialog(lessonFrame, "Are
you sure you want to clear?", "Confirm Clear", JOptionPane.YES_NO_OPTION);

                if(confirm == JOptionPane.YES_OPTION) {

                    subjectfield.setText(""); //get rid of all the text in the
fields
                    goalfield.setText("");
                    durationfield.setText("");
                    materialsarea.setText("");
                    planarea.setText("");
                    notesarea.setText("");

                    File file = new File(fileName); //will open the same file
                    try(BufferedWriter writer = new BufferedWriter(new
FileWriter(file))) {
                        writer.write("");  //clears the contents
                    }catch (IOException ex) {

JOptionPane.showMessageDialog(lessonFrame,JOptionPane.ERROR_MESSAGE);
                        return;
                    }
```

```
                    JOptionPane.showMessageDialog(lessonFrame,"Plan
cleared!","Success", JOptionPane.INFORMATION_MESSAGE);
                }

            }
        });

        load.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JFileChooser fileChooser = new JFileChooser();

                fileChooser.setCurrentDirectory(new File("templates")); //
current directory

                int filechoice = fileChooser.showOpenDialog(lessonFrame);

                boolean choice;
                if(filechoice==fileChooser.APPROVE_OPTION){
                    choice = true;
                }else{
                    choice=false;
                }
                if(choice){
                    File template=fileChooser.getSelectedFile();
                    try(BufferedReader reader = new BufferedReader(new
FileReader(template))){

                        //place all the text into the corresponding fields
                        //This is the same process as before
                        subjectfield.setText(reader.readLine());
                        goalfield.setText(reader.readLine());
                        durationfield.setText(reader.readLine());

                        reader.readLine();

                        StringBuilder materialscontent = new StringBuilder();
                        String line = reader.readLine();
                        while (line != null && !line.equals("===")) { //the
line must not be null and must not be "==="
                            materialscontent.append(line).append("\n");//adds
the string as a new line to the stringbuilder object
                            line = reader.readLine();//move to next line
                        }

                        line =reader.readLine(); //skip the "===" line

                        StringBuilder plancontent = new StringBuilder();
                        while (line != null && !line.equals("===")) {
```

```
                        plancontent.append(line).append("\n");
                        line = reader.readLine();
                    }

                    line = reader.readLine();

                    StringBuilder notescontent = new StringBuilder();
                    while (line != null) {
                        notescontent.append(line).append("\n");
                        line = reader.readLine();
                    }

                    materialsarea.setText(materialscontent.toString());
//now add the text
                    planarea.setText(plancontent.toString());
                    notesarea.setText(notescontent.toString());

                }catch (IOException ex) {
                    throw new RuntimeException(ex);
                }
            }
        }
    });
  }
}
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

public class PlanTemplate extends LessonPlan{
   private static JFrame currentLessonFrame;
   private JTextField filename;
   public PlanTemplate() {

       super("Template",0, null); //pass placeholder values; these don't
matter

       lessonFrame.setTitle("Lesson Plan Template"); //change the frame name

       load.setVisible(false); //hide load button; template page doesn't need
load functionality

       addfilename();

       ActionListener[] listeners = save.getActionListeners(); //action
listeners are stored as arrays
       for(int i=0;i< listeners.length;i++){ //I need to use a loop to delete
the lessonplan's save button listener
           save.removeActionListener(listeners[i]);
       }

       save.addActionListener(new ActionListener() {
           @Override
           public void actionPerformed(ActionEvent e) {
               savetemplate();
           }
       });

       lessonFrame.revalidate();
       lessonFrame.repaint();

   }

   private void addfilename(){ //create additional textfield for filename
       JPanel panel=new JPanel(new FlowLayout(FlowLayout.LEFT));
       JLabel label=new JLabel("File Name: ");
       filename = new JTextField(10);
       filename.setPreferredSize(new Dimension(100, 25));
       panel.add(label);
```

```
        panel.add(filename);

        // now I can add this panel to the parent JFrame
        // insert it at grid position (3, 0) spanning 3 columns
        GridBagConstraints gbc=new GridBagConstraints();
        gbc.insets=new Insets(5, 5, 5, 5);
        gbc.fill = GridBagConstraints.HORIZONTAL;
        gbc.gridx=3;
        gbc.gridy=0;
        gbc.gridwidth=3;
        lessonFrame.add(panel,gbc);
        lessonFrame.validate();

    }
        private void savetemplate(){
            String fileName = filename.getText();
            if (fileName.isEmpty()){ //do not allow empty filenames
                JOptionPane.showMessageDialog(lessonFrame,
                        "Please enter a template name.",
                        "Error",JOptionPane.ERROR_MESSAGE);
                return;
            }
            File dir=new File("templates"); //creates a directory
            if(dir.exists()==false){ //if there is a directory folder on the
computer
                dir.mkdir(); //method to create templates folder
            }
            File file = new File(dir,fileName+".txt");
            try (BufferedWriter writer=new BufferedWriter(new
FileWriter(file))) {
                writer.write(subjectfield.getText()); //write the file (same as
in lessonplan class)
                writer.newLine();
                writer.write(goalfield.getText());
                writer.newLine();
                writer.write(durationfield.getText());
                writer.newLine();

                //seperates the large text fields as these could be many lines
                writer.write("===");
                writer.newLine();
                // write the materials text
                writer.write(materialsarea.getText());
                writer.newLine();

                writer.write("===");
                writer.newLine();
                writer.write(planarea.getText());
                writer.newLine();
```

```
                writer.write("===");
                writer.newLine();
                writer.write(notesarea.getText());

            } catch (IOException ex) {
                JOptionPane.showMessageDialog(lessonFrame, "Error saving
template: " + ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
                return;
            }
            JOptionPane.showMessageDialog(lessonFrame, "Template saved
successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);
        }

}
```