Oppgave 1: Hilseknapp (Løsningsforslag)

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

class Hilseprogram {
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
        } catch (Exception e) {
            System.exit(1);
        }

        JFrame vindu = new JFrame("Hilseprogram");
        vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel panel = new JPanel();
        JLabel hilsen = new JLabel("");
        JButton knapp = new JButton("Hilseknapp");
        panel.add(hilsen); panel.add(knapp);
        vindu.add(panel);

        knapp.addActionListener(new ActionListener() {
            private boolean hilst = false;

            @Override
            public void actionPerformed(ActionEvent e) {
                if (!hilst) {
                    hilsen.setText("Hei paa deg!");
                    hilst = true;
                } else {
                    hilsen.setText("Ha det!");
                    hilst = false;
                }
            }
        });

        vindu.pack();
        vindu.setVisible(true);
        vindu.setLocationRelativeTo(null);
    }
}
```

Oppgave 2: Digital klokke (Løsningsforslag)

```
import java.awt.*; import java.awt.event.*; import javax.swing.*;
import java.time.LocalTime;

class Klokke {
    private static JLabel klokkeslett = new JLabel(naa());

    private static String naa(){
        //Henter den lokale tiden naa paa format "12:34:56"
        LocalTime t = LocalTime.now();
        return String.format("%02d:%02d:%02d",
            t.getHour(), t.getMinute(), t.getSecond());
    }

    private static class Sekundteller implements Runnable {
        @Override
        public void run() {
            boolean fortsett = true;
            while(fortsett) {
                try {
                    Thread.sleep(1000);
                } catch(InterruptedException e){
                    fortsett = false;
                }
                klokkeslett.setText(naa());
            }
        }
    }

    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
        } catch (Exception e) { System.exit(1); }

        JFrame vindu = new JFrame("Klokke");
        vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel panel = new JPanel();
        vindu.add(panel);
        panel.add(klokkeslett);

        vindu.pack();
        vindu.setLocationRelativeTo(null);
        vindu.setVisible(true)
        new Thread(new Sekundteller()).start();
    }
}
```

Oppgave 3: Digital klokke med start- og stoppknapp (Løsningsforslag)

```
import java.awt.*; import java.awt.event.*; import javax.swing.*;
import java.time.LocalTime;

class Klokke {
    private static JLabel klokkeslett = new JLabel(naa());
    private static boolean gaarKlokken = false;

    private static String naa() {
        LocalTime t = LocalTime.now();
        return String.format("%02d:%02d:%02d",
            t.getHour(), t.getMinute(), t.getSecond());
    }

    private static class Sekundteller implements Runnable {
        @Override
        public void run() {
            while(true) {
                try {
                    Thread.sleep(1000);
                } catch(InterruptedException e){
                    gaarKlokken = false;
                }
                if(gaarKlokken) {
                    klokkeslett.setText(naa());
                }
            }
        }
    }

    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
        } catch (Exception e) { System.exit(1); }

        JFrame vindu = new JFrame("Klokke");
        vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel panel = new JPanel(); JButton knapp = new JButton("Start/Stopp");
        panel.add(klokkeslett); panel.add(knapp); vindu.add(panel);

        knapp.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                gaarKlokken = !gaarKlokken;
            }
        });
```

```
        vindu.pack(); vindu.setLocationRelativeTo(null); vindu.setVisible(true);
        new Thread(new Sekundteller()).start();
    }
}
```

Oppgave 4: Tekstinput fra tastatur (Løsningsforslag)

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

public class Tekstinput {
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
        } catch (Exception e) { System.exit(1); }

        JFrame vindu = new JFrame("Tekst Input");
        vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        vindu.setSize(500, 100);

        JPanel panel = new JPanel();
        vindu.add(panel);

        JTextField tekstfelt = new JTextField(20);
        panel.add(tekstfelt);

        JLabel label = new JLabel();
        panel.add(label);

        tekstfelt.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String tekst = tekstfelt.getText();
                label.setText(tekst);
                tekstfelt.setText("");
            }
        });

        vindu.setLocationRelativeTo(null);
        vindu.setVisible(true);
    }
}
```

Oppgave 5: Filviser (Løsningsforslag)

```
import java.io.File; import java.io.FileNotFoundException;
import java.util.Scanner;
import javax.swing.*; import java.awt.*; import java.awt.event.*;

class FilLeser {
    private String streng = "";

    public FilLeser(File fil) throws FileNotFoundException {
        lesFraFil(fil);
    }

    private void lesFraFil(File fil) throws FileNotFoundException {
        Scanner sc = new Scanner(fil);
        streng += "<html>";
        while(sc.hasNextLine()){
            streng += sc.nextLine() + "<br/>";
        }
        streng += "</html>";
    }
    public String hentStreng(){
        return streng;
    }
}

public class TekstGUI {
    private static JLabel tekst = new JLabel("Velg en fil, saa skal jeg vise deg den:");

    private static class VelgFilAction implements ActionListener{
        @Override
        public void actionPerformed(ActionEvent e){
            JFileChooser velger = new JFileChooser();
            int resultat = velger.showOpenDialog(null);

            if(resultat == JFileChooser.APPROVE_OPTION){
                try{
                    FilLeser leser = new FilLeser(velger.getSelectedFile());
                    System.out.println(leser.hentStreng());
                    tekst.setText(leser.hentStreng());
                } catch(FileNotFoundException error){
                    tekst.setText("Noe gikk galt, prov paa nytt");
                }
            } else {
                tekst.setText("Noe gikk galt, prov paa nytt");
            }
        }
    }
```

```
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
        } catch (Exception e) { System.exit(1); }

        JFrame vindu = new JFrame("Filvelger");
        vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel panel = new JPanel(); panel.add(tekst);
        vindu.add(panel);

        JButton velgFilKnapp = new JButton("Velg fil");
        velgFilKnapp.addActionListener(new VelgFilAction());
        panel.add(velgFilKnapp);

        vindu.pack(); vindu.setLocationRelativeTo(null); vindu.setVisible(true);
    }
}
```