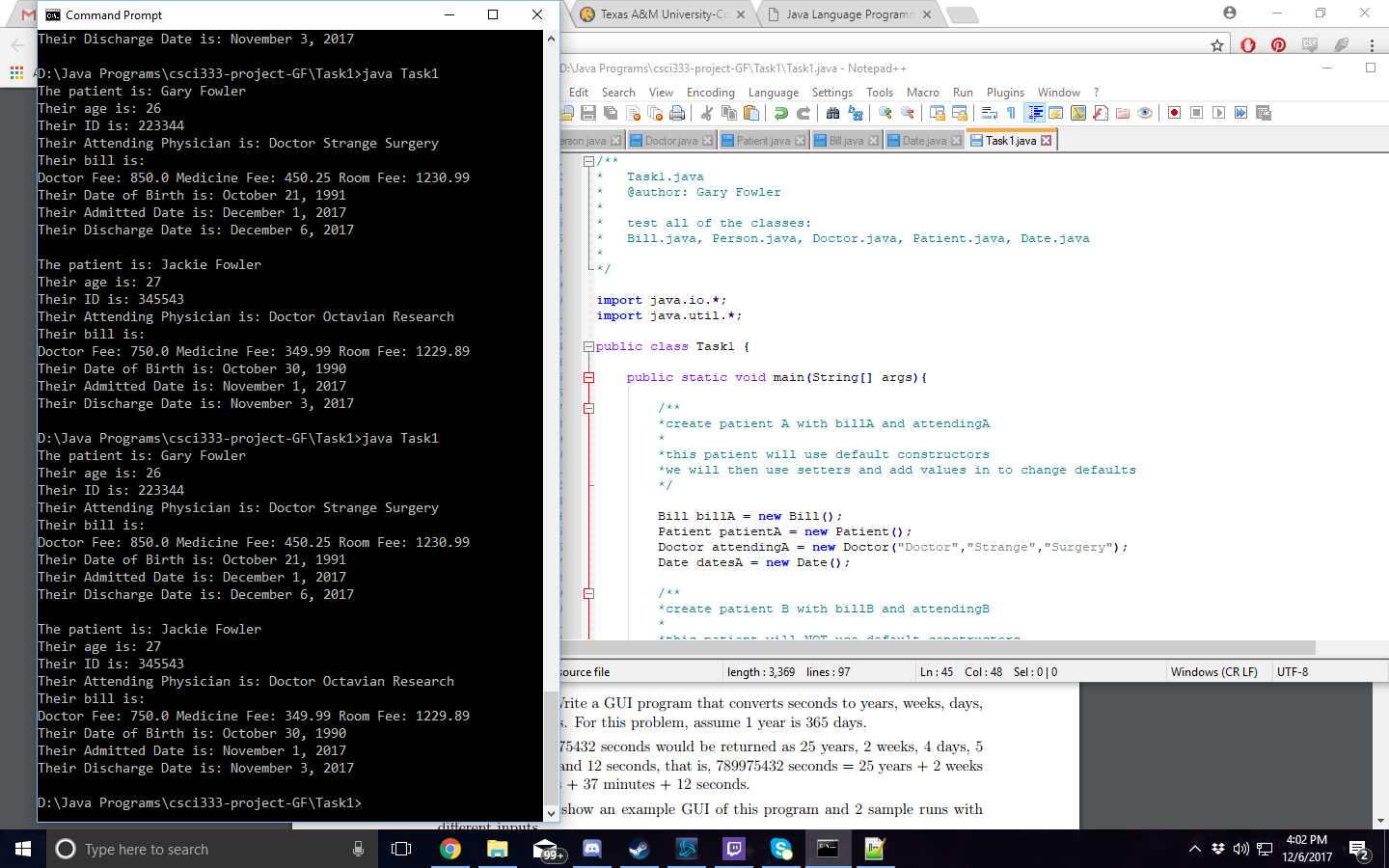
Output of Task1GF.java



Task1 source code

/\*\*

\* Person.java

\* stores information for a person, is the parent class of doctor and patient

\*/

public class Person

{

public static String firstName; //store the first name

public static String lastName; //store the last name

//Default constructor

//Initialize firstName and lastName to an empty string.

//Postcondition: firstName = ""; lastName = "";

public Person()

{

firstName = "";

lastName = "";

}

//Constructor with parameters

//Set firstName and lastName according to the parameters.

//Postcondition: firstName = first; lastName = last;

public Person(String first, String last)

{

setName(first, last);

}

//Method to output the first name and last name

//in the form firstName lastName.

public String toString()

{

return (firstName + " " + lastName);

}

/\*\*

\*setters

\*/

//Method to set firstName and lastName according to

//the parameters.

//Postcondition: firstName = first; lastName = last;

public void setName(String first, String last)

{

firstName = first;

lastName = last;

}

/\*\*

\*getters

\*/

//Method to return firstName.

//Postcondition: The value of firstName is returned.

public String getFirstName()

{

return firstName;

}

//Method to return lastName.

//Postcondition: The value of lastName is returned.

public String getLastName()

{

return lastName;

}

}

/\*\*

\* Doctor.java

\* @author: Gary Fowler

\* this is a subclass to Person, holds info for Doctor and his specialty

\*

\*/

public class Doctor extends Person {

//adds one field holding specialty

private String specialty = "";

//default Constructor

public Doctor(){

super(firstName, lastName);

specialty = "";

}

//Constructor with parameters

public Doctor(String first, String last, String spec){

super(first, last);

specialty = spec;

}

/\*\*

\*setters

\*/

//set name of doctor

public void setName(String first, String last){

firstName = first;

lastName = last;

}

//method to set the Doctor's specialty

public void setSpecialty(String spec){

specialty = spec;

}

/\*\*

\*getters

\*/

//method to return Doctor's specialty

public String getSpecialty(){

return specialty;

}

//method to get full doctor name

public String getName(){

String fullName = firstName + " " + lastName + " " + specialty;

return fullName;

}

}

/\*\*

\* Patient.java

\* @author: Gary Fowler

\* this is the subclass to Person, holds information for the Patient

\*

\*/

public class Patient extends Person {

private String ID;

private int age;

private String doctorInfo;

private String admittedDate;

private String dateOfBirth;

private String dischargeDate;

//default Constructor

public Patient(){

super(firstName, lastName);

age = 0;

ID = "123456"; //default ID, 6 digits

doctorInfo = "default"; //pull from Doctor.java

}

//Constructor with parameters

public Patient(String first, String last, int a, String id, String dname){

super(firstName, lastName);

setAge(a);

setID(id);

setDoctor(dname); //pull from Doctor.java

}

//method to output the Patient's information

public String toString(){

return (firstName + " " + lastName + " " + age + " " + ID + " " + doctorInfo);

}

/\*\*

\*setters

\*/

//set patients age

public void setAge(int a){

age = a;

}

//set patients id

public void setID(String id){

ID = id;

}

//set name of patient

public void setName(String first, String last){

firstName = first;

lastName = last;

}

//set the doctorName

public void setDoctor(String info){

doctorInfo = info;

}

//set the admitted date

public void setAdmitted(String ad){

admittedDate = ad;

}

//set the date of birth

public void setDOB(String dob){

dateOfBirth = dob;

}

//set the discharge date

public void setDischarge(String dd){

dischargeDate = dd;

}

/\*\*

\*getters

\*/

//Method to return firstName.

//Postcondition: The value of firstName is returned.

public String getFirstName()

{

return firstName;

}

//Method to return lastName.

//Postcondition: The value of lastName is returned.

public String getLastName()

{

return lastName;

}

//get patients age

public int getAge(){

return age;

}

//get patients id

public String getID(){

return ID;

}

//get doctor information

public String getDoctor(){

return doctorInfo;

}

//method to get full patient name

public String getName(){

String fullName = firstName + " " + lastName;

return fullName;

}

//get the admitted date

public String getAdmitted(){

return admittedDate;

}

//get the date of birth

public String getDOB(){

return dateOfBirth;

}

//get the discharge date

public String getDischarge(){

return dischargeDate;

}

}

/\*\*

\* Bill.java

\* @author: Gary Fowler

\* this class determines the patients bill and outputs that information

\*/

public class Bill {

private String ID;

private double totalCharges;

private double docFee;

private double medFee;

private double roomFee;

//Default constructor

//Initialize all values to 0, except the default 6 digit ID

public Bill(){

ID = "123456"; //ID is 6 digit

docFee = 0;

medFee = 0;

roomFee = 0;

totalCharges = docFee + medFee + roomFee;

}

//Constructor with parameters

//we take the three fees and set them and total charges

public Bill(String ID, double docFee, double medFee, double roomFee){

setID(ID);

setCharges(docFee, medFee, roomFee);

}

/\*\*

\*setters

\*/

//first set the ID

public void setID(String i){

ID = i;

}

//now we set not only the total charges but also all the individuals

public void setCharges(double doc, double med, double room){

totalCharges = doc + med + room;

docFee = doc;

medFee = med;

roomFee = room;

}

/\*\*

\*getters

\*/

//return the ID

public String getID(){

return ID;

}

//return the totalCharges

public double getTotalCharges(){

return totalCharges;

}

//return each individual fee

//doc

public double getDocFee(){

return docFee;

}

//med

public double getMedFee(){

return medFee;

}

//room

public double getRoomFee(){

return roomFee;

}

}

/\*\*

\* Date.java

\* @author: Gary Fowler

\* this class holds information on important patient dates

\*/

public class Date {

private String dateOfBirth;

private String admittedDate;

private String dischargeDate;

//Default constructor

//Initialize all values to 0, except the default 6 digit ID

public Date(){

dateOfBirth = "default";

admittedDate = "default";

dischargeDate = "default";

}

//Constructor with parameters

//we take the three fees and set them and total charges

public Date(String dob, String ad, String dd){

setDOB(dob);

setAdmitted(ad);

setDischarge(dd);

}

/\*\*

\*setters

\*/

//set the necessary dates

public void setDates(String dob, String ad, String dd){

dateOfBirth = dob;

admittedDate = ad;

dischargeDate = dd;

}

//set the date of birth

public void setDOB(String dob){

dateOfBirth = dob;

}

//set the admitted date

public void setAdmitted(String ad){

admittedDate = ad;

}

//set the discharge date

public void setDischarge(String dd){

dischargeDate = dd;

}

/\*\*

\*getters

\*/

//return the date of birth

public String getDOB(){

return dateOfBirth;

}

//return the admitted date

public String getAdmitted(){

return admittedDate;

}

//return the discharge date

public String getDischarge(){

return dischargeDate;

}

}

/\*\*

\* Task1.java

\* @author: Gary Fowler

\*

\* test all of the classes:

\* Bill.java, Person.java, Doctor.java, Patient.java, Date.java

\*

\*/

import java.io.\*;

import java.util.\*;

public class Task1 {

public static void main(String[] args){

/\*\*

\*create patient A with billA and attendingA

\*

\*this patient will use default constructors

\*we will then use setters and add values in to change defaults

\*/

Bill billA = new Bill();

Patient patientA = new Patient();

Doctor attendingA = new Doctor("Doctor","Strange","Surgery");

Date datesA = new Date();

/\*\*

\*create patient B with billB and attendingB

\*

\*this patient will NOT use default constructors

\*we will then only use getters to output their information

\*/

Bill billB = new Bill("345543",750.00,349.99,1229.89);

Doctor attendingB = new Doctor("Doctor","Octavian","Research");

Patient patientB = new Patient("Jackie","Fowler",27,"345543",attendingB.getName());

Date datesB = new Date("October 30, 1990","November 1, 2017", "November 3, 2017");

/\*\*

\* Patient A setters

\*/

//doctor setters

attendingA.setName("Doctor","Strange");

attendingA.setSpecialty("Surgery");

//patient setters

patientA.setAge(26);

patientA.setID("223344");

patientA.setDoctor(attendingA.getName()); //pull this from Doctor.java

patientA.setName("Gary","Fowler");

//bill setters

billA.setID(patientA.getID()); //we actually want to get the ID from Patient.java

billA.setCharges(850.00,450.25,1230.99);

//date setters

datesA.setDates("October 21, 1991","December 1, 2017","December 6, 2017");

patientA.setDOB(datesA.getDOB());

patientA.setAdmitted(datesA.getAdmitted());

patientA.setDischarge(datesA.getDischarge());

/\*\*

\* Patient A output

\*/

System.out.println("The patient is: " + patientA.getName());

System.out.println("Their age is: " + patientA.getAge());

System.out.println("Their ID is: " + patientA.getID());

System.out.println("Their Attending Physician is: " + patientA.getDoctor());

System.out.println("Their bill is: ");

System.out.print("Doctor Fee: " + billA.getDocFee() + " ");

System.out.print("Medicine Fee: " + billA.getMedFee() + " ");

System.out.println("Room Fee: " + billA.getRoomFee());

System.out.println("Their Date of Birth is: " + patientA.getDOB());

System.out.println("Their Admitted Date is: " + patientA.getAdmitted());

System.out.println("Their Discharge Date is: " + patientA.getDischarge());

/\*\*

\* Patient B output

\*/

patientB.setName("Jackie","Fowler");

System.out.println("\nThe patient is: " + patientB.getName());

System.out.println("Their age is: " + patientB.getAge());

System.out.println("Their ID is: " + patientB.getID());

System.out.println("Their Attending Physician is: " + patientB.getDoctor());

System.out.println("Their bill is: ");

System.out.print("Doctor Fee: " + billB.getDocFee() + " ");

System.out.print("Medicine Fee: " + billB.getMedFee() + " ");

System.out.println("Room Fee: " + billB.getRoomFee());

System.out.println("Their Date of Birth is: " + datesB.getDOB());

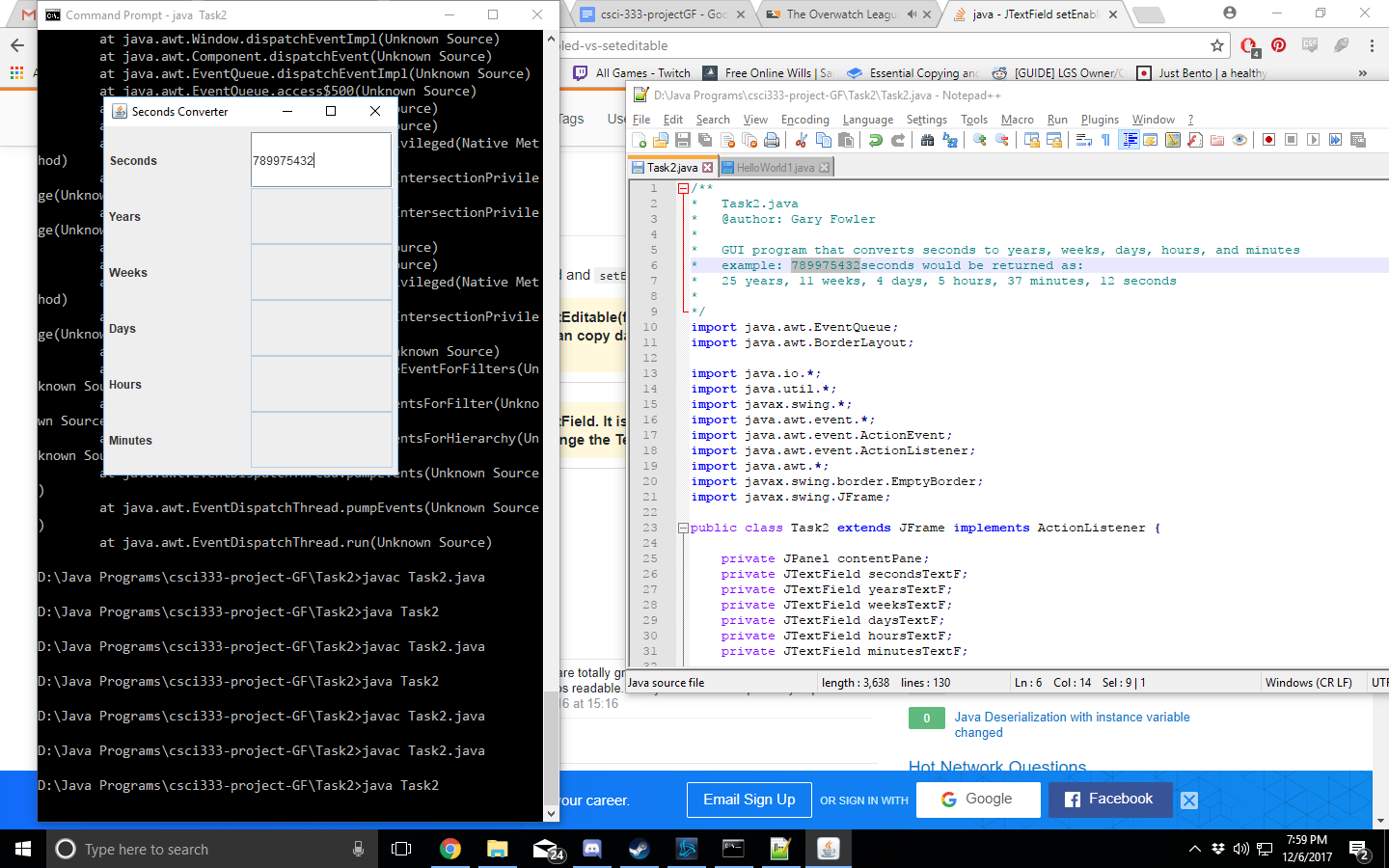
System.out.println("Their Admitted Date is: " + datesB.getAdmitted());

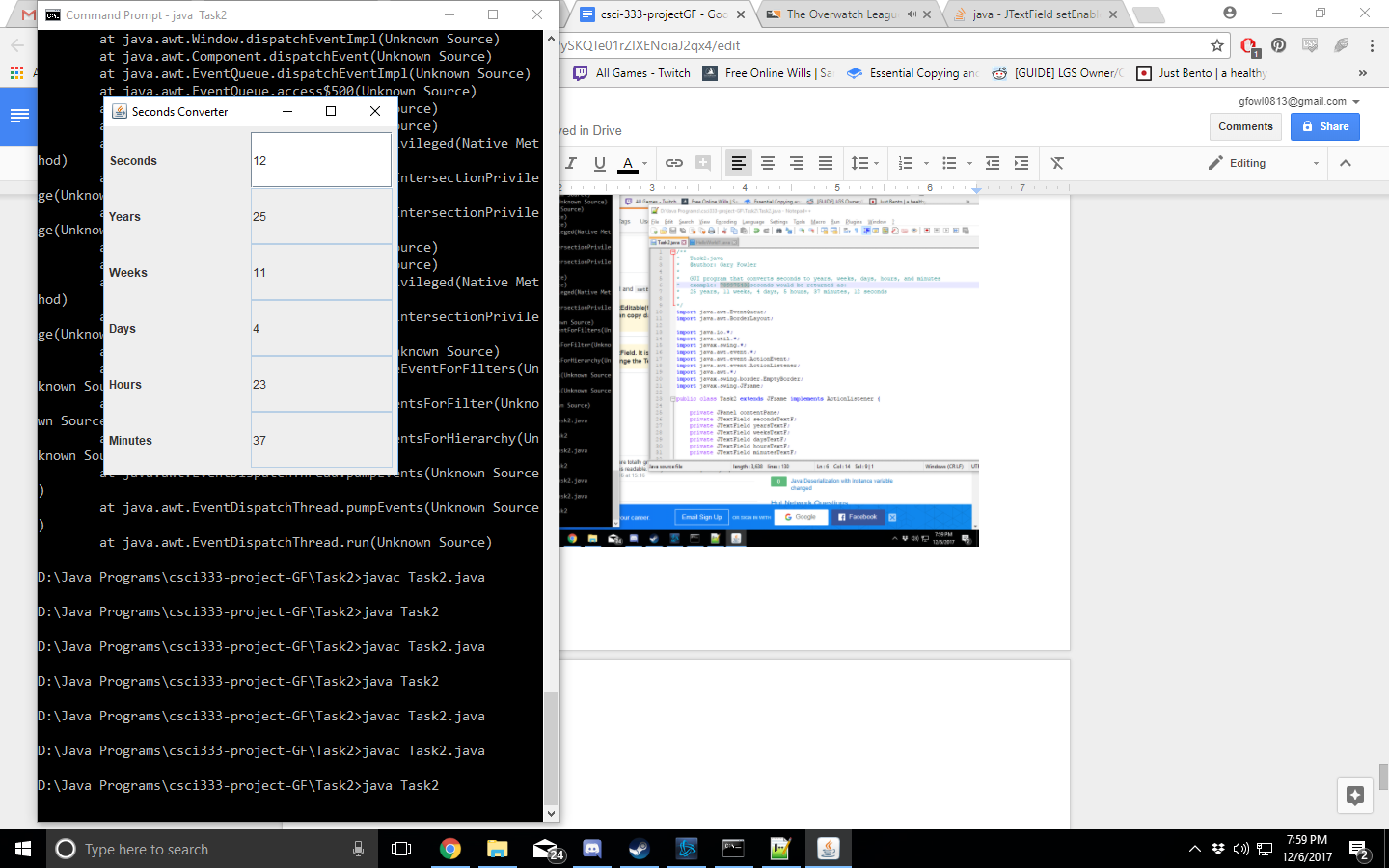
System.out.println("Their Discharge Date is: " + datesB.getDischarge());

}

}

Task2 Output





Source code

/\*\*

\* Task2.java

\* @author: Gary Fowler

\*

\* GUI program that converts seconds to years, weeks, days, hours, and minutes

\* example: 789975432seconds would be returned as:

\* 25 years, 11 weeks, 4 days, 5 hours, 37 minutes, 12 seconds

\*

\*/

import java.awt.EventQueue;

import java.awt.BorderLayout;

import java.io.\*;

import java.util.\*;

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.\*;

import javax.swing.border.EmptyBorder;

import javax.swing.JFrame;

public class Task2 extends JFrame implements ActionListener {

private JPanel contentPane;

private JTextField secondsTextF;

private JTextField yearsTextF;

private JTextField weeksTextF;

private JTextField daysTextF;

private JTextField hoursTextF;

private JTextField minutesTextF;

private JLabel secondsLabel, yearsLabel, weeksLabel,

daysLabel, hoursLabel, minutesLabel;

//the chosen layout

GridLayout converterLayout = new GridLayout(0,2);

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Task2 frame = new Task2();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public Task2(){

/\*\*

\*outer container

\*/

setTitle("Seconds Converter");

setBounds(100,100,320,400);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

//set our layout

contentPane.setLayout(converterLayout);

/\*\*

\*add elements to the gridlayout

\*/

contentPane.add(new JLabel("Seconds"));

secondsTextF = new JTextField();

secondsTextF.addActionListener(this);

contentPane.add(secondsTextF);

contentPane.add(new JLabel("Years"));

yearsTextF = new JTextField();

contentPane.add(yearsTextF);

yearsTextF.setEditable(false);

contentPane.add(new JLabel("Weeks"));

weeksTextF = new JTextField();

contentPane.add(weeksTextF);

weeksTextF.setEditable(false);

contentPane.add(new JLabel("Days"));

daysTextF = new JTextField();

contentPane.add(daysTextF);

daysTextF.setEditable(false);

contentPane.add(new JLabel("Hours"));

hoursTextF = new JTextField();

contentPane.add(hoursTextF);

hoursTextF.setEditable(false);

contentPane.add(new JLabel("Minutes"));

minutesTextF = new JTextField();

contentPane.add(minutesTextF);

minutesTextF.setEditable(false);

}

//our conversion function and listener function

public void actionPerformed(ActionEvent arg0){

if(arg0.getSource()==secondsTextF){

String text = secondsTextF.getText();

int seconds = Integer.parseInt(text);

//convert to the corresponding values

int years = seconds / 31557600;

int weeks = (seconds % 31557600) / 86400;

int days = ((seconds % 31557600) % 604800) / 86400;

int hours = (((seconds % 31557600) % 604800) % 86400) / 3600;

int minutes = ((((seconds % 31557600) % 604800) % 86400) % 3600) / 60;

seconds = ((((seconds % 31557600) % 604800) % 86400) % 3600) % 60;

//set the text in the text boxes

secondsTextF.setText(Integer.toString(seconds));

yearsTextF.setText(Integer.toString(years));

weeksTextF.setText(Integer.toString(weeks));

daysTextF.setText(Integer.toString(days));

hoursTextF.setText(Integer.toString(hours));

minutesTextF.setText(Integer.toString(minutes));

}

}

}