**CSC 285**

**Session 9**

**Putting it all together (Part 4)**

**=============================================================================================**

**// User.java**

**import java.text.DecimalFormat;**

**public class User implements UserInterface{**

**//create variables**

**private int id;**

**private String name;**

**private char hourly;**

**private double wage;**

**private double avgHours;**

**private double percentRent;**

**private double foodBudget;**

**private double expenses;**

**//constructor**

**public User(int id, String name, char hourly, double wage, double avgHours, double percentRent, double foodBudget, double expenses){**

**//"this" is a reference to the object**

**this.id = id;**

**this.name = name;**

**this.hourly = hourly;**

**this.wage = wage;**

**this.avgHours = avgHours;**

**this.percentRent = percentRent;**

**this.foodBudget = foodBudget;**

**this.expenses = expenses;**

**}**

**//interface method implementation**

**public double calc\_gross\_pay(char hourly, double wage, double avgHours){**

**//catch arguments**

**char tempHourly = hourly;**

**double tempWage = wage;**

**double tempAvgHours = avgHours;**

**//calculate gross monthly income**

**double grossPay;**

**if (tempHourly == 'y' || tempHourly == 'Y'){**

**grossPay = (((tempWage \* tempAvgHours) \* this.WEEKS) / this.MONTHS);**

**} else{**

**grossPay = (tempWage / this.MONTHS);**

**}**

**return (grossPay);**

**}//end calc\_gross\_pay**

**//interface method implementation**

**public double calc\_net\_pay(double grossPay){**

**//catch arguments**

**double tempGrossPay = grossPay;**

**double taxMod = (1.0 - this.TAX\_RATE);**

**return (tempGrossPay \* taxMod);**

**}//end calc\_net\_pay**

**//interface method implementation**

**public double calc\_budget(double netPay, double percentRent, double foodBudget, double expenses){**

**//catch arguments**

**double tempNetPay = netPay;**

**double tempPercentRent = percentRent;**

**double tempFoodBudget = foodBudget;**

**double tempExpenses = expenses;**

**//calculate budget**

**double budget = tempNetPay; //budget starts with net income**

**budget = (budget - (budget \* (tempPercentRent / 100))); //take rent off the top**

**budget -= (tempFoodBudget + tempExpenses); //then subtract other expenses**

**return (budget);**

**}//end calc\_budget**

**//interface method implementation**

**public double calc\_rent(double netPay, double percentRent){**

**//catch arguments**

**double tempNetPay = netPay;**

**double tempPercentRent = percentRent;**

**//calculate rent**

**double rent = (tempNetPay \* (tempPercentRent / 100)); //returns monthly budget for rent**

**return (rent);**

**}//end calc\_rent**

**public String toString()**

**{**

**DecimalFormat twoDigits = new DecimalFormat( "0.00" );**

**DecimalFormat dollars = new DecimalFormat( "$.00" );**

**return(**

**"\n Id: " + Integer.toString(this.id) +**

**"\n Name: " + this.name +**

**"\n Hourly Worker: " + this.hourly +**

**"\n Wage/Salary: " + dollars.format(this.wage) +**

**"\n Average Hours Per Week: " + twoDigits.format(this.avgHours) +**

**"\n Rent (Percent of Income): " + Double.toString(this.percentRent) + "%" +**

**"\n Rent (Monthly Budget): " + dollars.format(calc\_rent(calc\_net\_pay(calc\_gross\_pay(this.hourly, this.wage, this.avgHours)), this.percentRent)) +**

**"\n Food Budget: " + dollars.format(this.foodBudget) +**

**"\n Other Expenses: " + dollars.format(this.expenses) +**

**"\n Gross Income: " + dollars.format(calc\_gross\_pay(this.hourly, this.wage, this.avgHours)) +**

**"\n Net Income: " + dollars.format(calc\_net\_pay(calc\_gross\_pay(this.hourly, this.wage, this.avgHours))) +**

**"\n Budget: " + dollars.format(calc\_budget(calc\_net\_pay(calc\_gross\_pay(this.hourly, this.wage, this.avgHours)), this.percentRent, this.foodBudget, this.expenses)));**

**}//end toString**

**}**

**//======================================================================================**

**//UserRecord.java**

**import java.io.\*;**

**import java.io.RandomAccessFile;**

**public class UserRecord{**

**//create variables**

**private int id;**

**private String name;**

**private char hourly;**

**private double wage;**

**private double avgHours;**

**private double percentRent;**

**private double foodBudget;**

**private double expenses;**

**//default constructor**

**public UserRecord(){**

**id = 0;**

**name = "";**

**hourly = ' ';**

**wage = 0.0;**

**avgHours = 0.0;**

**percentRent = 0.0;**

**foodBudget = 0.0;**

**expenses = 0.0;**

**}**

**//write to a file**

**public void write(RandomAccessFile file) throws IOException{**

**//catch arguments**

**RandomAccessFile tempFile = file;**

**//write to the file**

**tempFile.writeInt(id);// 4 bytes**

**StringBuffer buffer; //using string buffer beacuse it is multithreaded**

**if(name != null){**

**buffer = new StringBuffer(name);**

**}else{**

**buffer = new StringBuffer(12);**

**}**

**buffer.setLength(12); //using a set length so the byte size is known**

**tempFile.writeChars(buffer.toString()); //write string as sequense of characters (12\*2 = 24 bytes)**

**tempFile.writeChar(hourly);// 2 bytes**

**tempFile.writeDouble(wage);// 8 bytes**

**tempFile.writeDouble(avgHours);// 8 bytes**

**tempFile.writeDouble(percentRent);// 8 bytes**

**tempFile.writeDouble(foodBudget);// 8 bytes**

**tempFile.writeDouble(expenses);// 8 bytes**

**}//end write (4 + 24 + 2 + 8 + 8 + 8 + 8 + 8) = 70 bytes**

**//read from a file**

**public void read(RandomAccessFile file) throws IOException{**

**//catch arguments**

**RandomAccessFile tempFile = file;**

**id = tempFile.readInt();**

**char[] charArray = new char[12];**

**for(int i=0; i < charArray.length; i++){**

**charArray[i] = tempFile.readChar();**

**}**

**name = new String(charArray); //retrieves and writes to string next 12 chars from file**

**hourly = tempFile.readChar();**

**wage = tempFile.readDouble();**

**avgHours = tempFile.readDouble();**

**percentRent = tempFile.readDouble();**

**foodBudget = tempFile.readDouble();**

**expenses = tempFile.readDouble();**

**}//end read**

**//getters and setters**

**public int getId(){**

**return this.id;**

**}**

**public void setId(int id){**

**this.id = id;**

**}**

**public String getName(){**

**return this.name;**

**}**

**public void setName(String name){**

**this.name = name;**

**}**

**public char getHourly(){**

**return this.hourly;**

**}**

**public void setHourly(char hourly){**

**this.hourly = hourly;**

**}**

**public double getWage(){**

**return this.wage;**

**}**

**public void setWage(double wage){**

**this.wage = wage;**

**}**

**public double getAvgHours(){**

**return this.avgHours;**

**}**

**public void setAvgHours(double avgHours){**

**this.avgHours = avgHours;**

**}**

**public double getPercentRent(){**

**return this.percentRent;**

**}**

**public void setPercentRent(double percentRent){**

**this.percentRent = percentRent;**

**}**

**public double getFoodBudget(){**

**return this.foodBudget;**

**}**

**public void setFoodBudget(double foodBudget){**

**this.foodBudget = foodBudget;**

**}**

**public double getExpenses(){**

**return this.expenses;**

**}**

**public void setExpenses(double expenses){**

**this.expenses = expenses;**

**}**

**//returns size in bytes of record**

**public static int size(){**

**return 70; //(4 + 24 + 2 + 8 + 8 + 8 + 8 + 8) = 70 bytes**

**}**

**}**

**//=================================================================================**

**//UserInterface.java**

**public interface UserInterface{**

**//Constants (tax, week and month intervals)**

**public double TAX\_RATE = 0.1925;**

**public int WEEKS = 52;**

**public int MONTHS = 12;**

**//methods to be implemented by User object**

**public double calc\_gross\_pay(char hourly, double avgHours, double wage);**

**public double calc\_net\_pay(double grossPay);**

**public double calc\_budget(double netPay, double percentRent, double foodBudget, double expenses);**

**public double calc\_rent(double netPay, double percentRent);**

**}**

**//==============================================================================**

**//WriteUserRecord.java**

**import javafx.application.Application;**

**import javafx.scene.Scene;**

**import javafx.scene.control.Button;**

**import javafx.stage.Stage;**

**import javafx.scene.control.Label;**

**import javafx.scene.control.TextField;**

**import javafx.scene.control.RadioButton;**

**import javafx.scene.control.ToggleGroup;**

**import javafx.scene.control.Spinner;**

**import javafx.scene.control.SpinnerValueFactory;**

**import javafx.scene.control.SpinnerValueFactory.DoubleSpinnerValueFactory;**

**import javafx.scene.layout.VBox;**

**import javafx.scene.layout.HBox;**

**import javafx.scene.layout.GridPane;**

**import javafx.geometry.Pos;**

**import javafx.geometry.HPos;**

**import javafx.geometry.Insets;**

**import javafx.event.ActionEvent;**

**import javafx.event.EventHandler;**

**import java.io.RandomAccessFile;**

**import java.io.\*;**

**public class WriteUserRecord extends Application{**

**//create fields**

**private TextField idField, nameField, wageField, avgHoursField, foodBudgetField, expensesField;**

**private ToggleGroup hourlyOptions;**

**private HBox hourlyBox;**

**private Spinner percentRentSpinner;**

**//create buttons**

**private RadioButton yesOption, noOption;**

**private Button done, next;**

**private RandomAccessFile outputFile;**

**private UserRecord record;**

**//override start()**

**public void start(Stage primaryStage){**

**record = new UserRecord();//create a new record**

**//open a file**

**try{**

**outputFile = new RandomAccessFile("userRecord.dat", "rw"); //rw means read and write**

**}catch(IOException er){**

**System.err.println("error opening file: " + er.toString());**

**System.exit(1);**

**}**

**//create a pane**

**GridPane pane = new GridPane();**

**pane.setAlignment(Pos.CENTER);**

**pane.setHgap(5);//set horizontal spacing**

**pane.setVgap(10);//set vertical spacing**

**//set border padding**

**pane.setPadding(new Insets(5, 5, 5, 5));**

**//create form**

**pane.add(new Label("Id: "), 0, 0); //over, down**

**idField = new TextField();**

**pane.add(idField, 1, 0); //positions input field to the right of label**

**pane.add(new Label("Name: "), 0, 1);**

**nameField = new TextField();**

**pane.add(nameField, 1, 1);**

**pane.add(new Label("Hourly Worker: "), 0, 2);**

**hourlyOptions = new ToggleGroup(); //create a group for radio buttons**

**yesOption = new RadioButton("Yes");**

**yesOption.setToggleGroup(hourlyOptions);**

**yesOption.setSelected(true);//sets default option to "yes"**

**//pane.add(yesOption, 1, 2); //add yes button to group and add to scene**

**noOption = new RadioButton("No");**

**noOption.setToggleGroup(hourlyOptions);**

**//pane.add(noOption, 2, 2);//add no button to group and add to scene**

**hourlyBox = new HBox(20.0, yesOption, noOption);**

**pane.add(hourlyBox, 1, 2);//add radio buttons to the scene**

**pane.add(new Label("Wage/Salary: "), 0, 3);**

**wageField = new TextField();**

**pane.add(wageField, 1, 3);**

**pane.add(new Label("Average Hours Worked Per Week: "), 0, 4);**

**avgHoursField = new TextField();**

**pane.add(avgHoursField, 1, 4);**

**pane.add(new Label("Rent (Percent of Income): "), 0, 5);**

**percentRentSpinner = new Spinner(10.0, 50.0, 30.0, 5.0); //Spinner(double minValue, double maxValue, double startValue, double stepValue)**

**pane.add(percentRentSpinner, 1, 5);**

**pane.add(new Label("Food Budget: "), 0, 6);**

**foodBudgetField = new TextField();**

**pane.add(foodBudgetField, 1, 6);**

**pane.add(new Label("Other Expenses: "), 0, 7);**

**expensesField = new TextField();**

**pane.add(expensesField, 1, 7);**

**next = new Button("Next");**

**next.setMaxWidth(Double.MAX\_VALUE);**

**NextButton handler1 = new NextButton();//create handler**

**next.setOnAction(handler1); //register handler**

**pane.add(next, 0, 8);**

**GridPane.setHalignment(next, HPos.LEFT);**

**done = new Button("Done");**

**done.setMaxWidth(Double.MAX\_VALUE);**

**DoneButton handler2 = new DoneButton(); //create handler**

**done.setOnAction(handler2);//register handler**

**pane.add(done, 1, 8);**

**GridPane.setHalignment(done, HPos.RIGHT);**

**Scene scene = new Scene(pane);**

**primaryStage.setTitle("Glen CIT-285 Final - Write Record");**

**primaryStage.setWidth(400);**

**primaryStage.setHeight(400);**

**primaryStage.setScene(scene);**

**primaryStage.show();**

**}//end start**

**void addRecord(){**

**//create "stunt" variables**

**int tempId = 0;**

**char tempHourly;**

**double tempWage, tempAvgHours, tempPercentRent, tempFoodBudget, tempExpenses;**

**if(!idField.getText().equals("")){//user must input an id**

**try{**

**tempId = Integer.parseInt(idField.getText());**

**if(tempId >= 1){**

**record.setId(tempId); //set user id**

**}**

**}catch(NumberFormatException nfe){**

**System.err.println("Id number must be entered as an Integer");**

**}**

**record.setName(nameField.getText());//set user name**

**if (hourlyOptions.getSelectedToggle() == yesOption){**

**tempHourly = 'Y';**

**} else {**

**tempHourly = 'N';**

**}**

**record.setHourly(tempHourly);// set user hourly status**

**try{**

**tempWage = new Double(wageField.getText());**

**record.setWage(tempWage);//set wage/salary**

**}catch(NumberFormatException nfe){**

**System.err.println("Number must be entered as Double");**

**}**

**try{**

**tempAvgHours = new Double(avgHoursField.getText());**

**record.setAvgHours(tempAvgHours);// set average hours**

**}catch(NumberFormatException nfe){**

**System.err.println("Number must be entered as a Double");**

**}**

**tempPercentRent = new Double(percentRentSpinner.getValue().toString());**

**record.setPercentRent(tempPercentRent);//set rent percentage**

**try{**

**tempFoodBudget = new Double(foodBudgetField.getText());**

**record.setFoodBudget(tempFoodBudget);// set food budget**

**}catch(NumberFormatException nfe){**

**System.err.println("Number must be entered as a Double");**

**}**

**try{**

**tempExpenses = new Double(expensesField.getText());**

**record.setExpenses(tempExpenses);// set expenses**

**}catch(NumberFormatException nfe){**

**System.err.println("Number must be entered as a Double");**

**}**

**//write information to a file**

**try{**

**outputFile.seek((long)(tempId-1) \* record.size());//set pointer**

**record.write(outputFile);**

**}catch(IOException er){**

**System.err.println("Error during write to a file: " + er.toString());**

**System.exit(1);**

**}**

**}//end if statement**

**//reset fields**

**idField.setText("");**

**nameField.setText("");**

**yesOption.setSelected(true);**

**wageField.setText("");**

**avgHoursField.setText("");**

**percentRentSpinner.setValueFactory(new DoubleSpinnerValueFactory(10.0, 50.0, 30.0, 5.0));**

**foodBudgetField.setText("");**

**expensesField.setText("");**

**}//end addRecord**

**class NextButton implements EventHandler<ActionEvent>{**

**public void handle(ActionEvent e){**

**addRecord();**

**}**

**}//end NextButton**

**class DoneButton implements EventHandler<ActionEvent>{**

**public void handle(ActionEvent e){**

**try{**

**outputFile.close(); //close a file**

**}catch(IOException er){**

**System.err.println("File not closed properly: " + er.toString());**

**System.exit(1);**

**}**

**System.exit(0);**

**}**

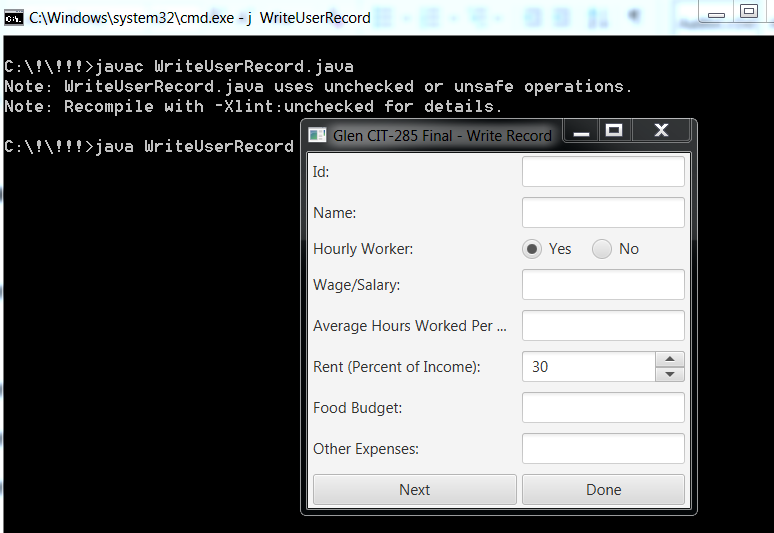
**}//end DoneButton**

**public static void main(String[] args){**

**launch(args);**

**}**

**}//end WriteUserRecord**



**//========================================================================================**

**//ReadUserRecord.java**

**import javafx.application.Application;**

**import javafx.scene.Scene;**

**import javafx.scene.control.Button;**

**import javafx.stage.Stage;**

**import javafx.scene.control.Label;**

**import javafx.scene.control.TextField;**

**import javafx.scene.control.RadioButton;**

**import javafx.scene.control.ToggleGroup;**

**import javafx.scene.control.Spinner;**

**import javafx.scene.control.SpinnerValueFactory;**

**import javafx.scene.control.SpinnerValueFactory.DoubleSpinnerValueFactory;**

**import javafx.scene.layout.VBox;**

**import javafx.scene.layout.HBox;**

**import javafx.scene.layout.GridPane;**

**import javafx.scene.layout.Background;**

**import javafx.scene.layout.BackgroundFill;**

**import javafx.scene.paint.Paint;**

**import javafx.scene.paint.Color;**

**import javafx.geometry.Pos;**

**import javafx.geometry.HPos;**

**import javafx.geometry.Insets;**

**import javafx.event.ActionEvent;**

**import javafx.event.EventHandler;**

**import java.io.RandomAccessFile;**

**import java.io.\*;**

**import java.text.DecimalFormat;**

**public class ReadUserRecord extends Application{**

**//create fields**

**private TextField idField, nameField, wageField, avgHoursField, foodBudgetField, expensesField, ammountRentField, grossPayField, netPayField, budgetField;**

**private ToggleGroup hourlyOptions;**

**private HBox hourlyBox;**

**private Spinner percentRentSpinner;**

**//create buttons**

**private RadioButton yesOption, noOption;**

**private Button done, next;**

**private RandomAccessFile inputFile;**

**private UserRecord record;**

**private User user;**

**//override start()**

**public void start(Stage primaryStage){**

**record = new UserRecord();//create a new record**

**//open a file**

**try{**

**inputFile = new RandomAccessFile("userRecord.dat", "rw"); //rw means read and write**

**}catch(IOException er){**

**System.err.println("error opening file: " + er.toString());**

**System.exit(1);**

**}**

**//create a pane**

**GridPane pane = new GridPane();**

**pane.setAlignment(Pos.CENTER);**

**pane.setHgap(5);//set horizontal spacing**

**pane.setVgap(10);//set vertical spacing**

**//set border padding**

**pane.setPadding(new Insets(5, 5, 5, 5));**

**//create form**

**pane.add(new Label("Id: "), 0, 0); //over, down**

**idField = new TextField();**

**idField.setEditable(false);**

**idField.setBackground(new Background(new BackgroundFill(Color.color(0.75,0.5,0,0.25), null, null)));**

**pane.add(idField, 1, 0); //positions input field to the right of label**

**pane.add(new Label("Name: "), 0, 1);**

**nameField = new TextField();**

**nameField.setEditable(false);**

**nameField.setBackground(new Background(new BackgroundFill(Color.color(0.75,0.5,0,0.25), null, null)));**

**pane.add(nameField, 1, 1);**

**pane.add(new Label("Hourly Worker: "), 0, 2);**

**hourlyOptions = new ToggleGroup(); //create a group for radio buttons**

**yesOption = new RadioButton("Yes");**

**yesOption.setToggleGroup(hourlyOptions);**

**yesOption.setSelected(true);//sets default option to "yes"**

**//yesOption.setDisabled(true);**

**//pane.add(yesOption, 1, 2); //add yes button to group and add to scene**

**noOption = new RadioButton("No");**

**//noOption.setDisabled(true);**

**noOption.setToggleGroup(hourlyOptions);**

**//pane.add(noOption, 2, 2);//add no button to group and add to scene**

**hourlyBox = new HBox(20.0, yesOption, noOption);**

**pane.add(hourlyBox, 1, 2);//add radio buttons to the scene**

**pane.add(new Label("Wage/Salary: "), 0, 3);**

**wageField = new TextField();**

**wageField.setEditable(false);**

**wageField.setBackground(new Background(new BackgroundFill(Color.color(0,0.75,0,0.25), null, null)));**

**pane.add(wageField, 1, 3);**

**pane.add(new Label("Average Hours Worked Per Week: "), 0, 4);**

**avgHoursField = new TextField();**

**avgHoursField.setEditable(false);**

**avgHoursField.setBackground(new Background(new BackgroundFill(Color.color(0.75,0.5,0,0.25), null, null)));**

**pane.add(avgHoursField, 1, 4);**

**pane.add(new Label("Rent (Percent of Income): "), 0, 5);**

**percentRentSpinner = new Spinner(10.0, 50.0, 30.0, 5.0); //Spinner(double minValue, double maxValue, double startValue, double stepValue)**

**//percentRentSpinner.setBackground(new Background(new BackgroundFill(Color.color(0.75,0.5,0,0.25), null, null)));**

**//percentRentSpinner.setDisabled(true);**

**pane.add(percentRentSpinner, 1, 5);**

**pane.add(new Label("Rent (Monthly Budget): "), 0, 6);**

**ammountRentField = new TextField();**

**ammountRentField.setEditable(false);**

**ammountRentField.setBackground(new Background(new BackgroundFill(Color.color(0.75,0,0,0.25), null, null)));**

**pane.add(ammountRentField, 1, 6);**

**pane.add(new Label("Food Budget: "), 0, 7);**

**foodBudgetField = new TextField();**

**foodBudgetField.setEditable(false);**

**foodBudgetField.setBackground(new Background(new BackgroundFill(Color.color(0.75,0,0,0.25), null, null)));**

**pane.add(foodBudgetField, 1, 7);**

**pane.add(new Label("Other Expenses: "), 0, 8);**

**expensesField = new TextField();**

**expensesField.setEditable(false);**

**expensesField.setBackground(new Background(new BackgroundFill(Color.color(0.75,0,0,0.25), null, null)));**

**pane.add(expensesField, 1, 8);**

**pane.add(new Label("Gross Income: "), 0, 9);**

**grossPayField = new TextField();**

**grossPayField.setEditable(false);**

**grossPayField.setBackground(new Background(new BackgroundFill(Color.color(0,0.75,0,0.25), null, null)));**

**pane.add(grossPayField, 1, 9);**

**pane.add(new Label("Net Income: "), 0, 10);**

**netPayField = new TextField();**

**netPayField.setEditable(false);**

**netPayField.setBackground(new Background(new BackgroundFill(Color.color(0,0.75,0,0.25), null, null)));**

**pane.add(netPayField, 1, 10);**

**pane.add(new Label("Income After Expenses and Taxes: "), 0, 11);**

**budgetField = new TextField();**

**budgetField.setEditable(false);**

**budgetField.setBackground(new Background(new BackgroundFill(Color.color(0,0.75,0,0.25), null, null)));**

**pane.add(budgetField, 1, 11);**

**next = new Button("Next");**

**next.setMaxWidth(Double.MAX\_VALUE);**

**NextButton handler1 = new NextButton();//create handler**

**next.setOnAction(handler1); //register handler**

**pane.add(next, 0, 12);**

**GridPane.setHalignment(next, HPos.LEFT);**

**done = new Button("Done");**

**done.setMaxWidth(Double.MAX\_VALUE);**

**DoneButton handler2 = new DoneButton(); //create handler**

**done.setOnAction(handler2);//register handler**

**pane.add(done, 1, 12);**

**GridPane.setHalignment(done, HPos.RIGHT);**

**Scene scene = new Scene(pane);**

**primaryStage.setTitle("Glen CIT-285 Final - Read Record");**

**primaryStage.setWidth(400);**

**primaryStage.setHeight(550);**

**primaryStage.setScene(scene);**

**primaryStage.show();**

**}//end start**

**void readRecord(){**

**DecimalFormat twoDigits = new DecimalFormat( "0.00" );**

**DecimalFormat dollars = new DecimalFormat( "$.00" );**

**//create "stunt" variables**

**int tempId = 0;**

**String tempName;**

**char tempHourly;**

**double tempWage, tempAvgHours, tempPercentRent, tempFoodBudget, tempExpenses;**

**try{**

**do{**

**record.read(inputFile);**

**tempId = record.getId();**

**tempName = record.getName();**

**tempHourly = record.getHourly();**

**tempWage = record.getWage();**

**tempAvgHours = record.getAvgHours();**

**tempPercentRent = record.getPercentRent();**

**tempFoodBudget = record.getFoodBudget();**

**tempExpenses = record.getExpenses();**

**user = new User(tempId, tempName, tempHourly, tempWage, tempAvgHours, tempPercentRent, tempFoodBudget, tempExpenses);**

**}while(record.getId() == 0);**

**System.out.println(user.toString());**

**idField.setText(String.valueOf(tempId));**

**nameField.setText(String.valueOf(tempName));**

**if (tempHourly == 'Y'){**

**yesOption.setSelected(true);**

**}else{**

**noOption.setSelected(true);**

**}**

**wageField.setText(String.valueOf(dollars.format(tempWage)));**

**avgHoursField.setText(String.valueOf(twoDigits.format(tempAvgHours)));**

**percentRentSpinner.setValueFactory(new DoubleSpinnerValueFactory(10.0, 50.0, tempPercentRent, 5.0));**

**ammountRentField.setText(String.valueOf(dollars.format(user.calc\_rent(user.calc\_net\_pay(user.calc\_gross\_pay(tempHourly, tempWage, tempAvgHours)), tempPercentRent))));**

**foodBudgetField.setText(String.valueOf(dollars.format(tempFoodBudget)));**

**expensesField.setText(String.valueOf(dollars.format(tempExpenses)));**

**grossPayField.setText(String.valueOf(dollars.format(user.calc\_gross\_pay(tempHourly, tempWage, tempAvgHours))));**

**netPayField.setText(String.valueOf(dollars.format(user.calc\_net\_pay(user.calc\_gross\_pay(tempHourly, tempWage, tempAvgHours)))));**

**budgetField.setText(String.valueOf(dollars.format(user.calc\_budget(user.calc\_net\_pay(user.calc\_gross\_pay(tempHourly, tempWage, tempAvgHours)), tempPercentRent, tempFoodBudget, tempExpenses))));**

**}catch(EOFException er){**

**closeFile();**

**}catch(IOException er){**

**System.err.println( "Error during read from file: " + er.toString() );**

**System.exit( 1 );**

**}**

**}//end readRecord**

**void closeFile(){**

**try{**

**inputFile.close(); //close a file**

**}catch(IOException er){**

**System.err.println("File not closed properly: " + er.toString());**

**System.exit(1);**

**}**

**System.exit(0);**

**}//end closeFile**

**class NextButton implements EventHandler<ActionEvent>{**

**public void handle(ActionEvent e){**

**readRecord();**

**}**

**}//end NextButton**

**class DoneButton implements EventHandler<ActionEvent>{**

**public void handle(ActionEvent e){**

**closeFile();**

**}**

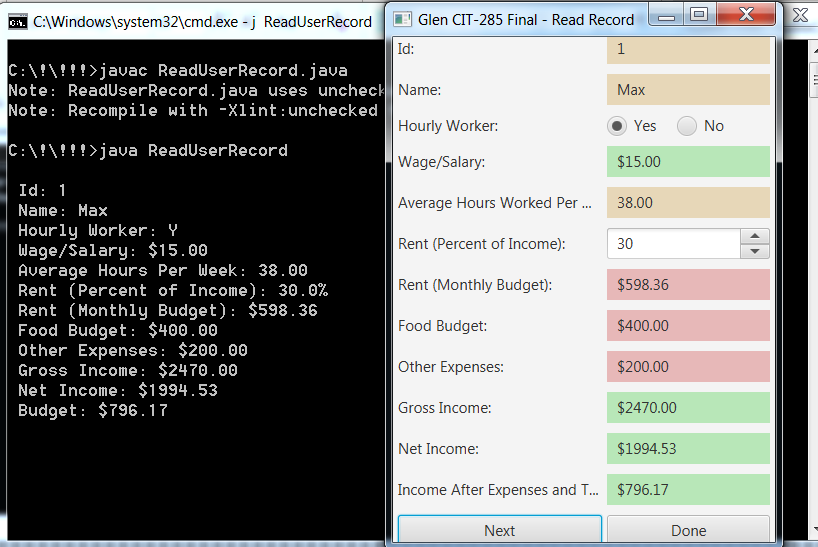
**}//end DoneButton**

**public static void main(String[] args){**

**launch(args);**

**}**

**}//end ReadUserRecord**



**C’est Finis**

**C’est Finis**