**CSC 285 Final Project Requirements**

**Here are the requirements for the Java Final Project!!!**

**First, here is the optional final project instructions!!!!**

**An optional final I mentioned is if you use the MySQL Series that I did and you add 2 fields to it I will accept this as a suitable final project.**

**I did a video how to add the two fields to the existing MySQL file for the EmployeeManager series that we did in class. Please note that you can also add the PreparedStatement to this series for significant credit!!!!**

**As a second final you had to add a phone number field to the Access Database. Now if you add 2 more fields and run it I will accept that as an alternate Final Project.**

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

**I will post a Final Project Templates that you may use. For reference and you will see how to add a Date using this template.**

**Finally, you may design your own unique final project.**

**You should not just copy my work or one of the designs offered in any of the templates that I have provided. If you do so, then you shall lose points!**

**No exceptions!**

**\*\*\*Please note that “YOU MAY NOT MAKE A GAME!!!” I WILL NOT ACCEPT ANY FINAL PROJECTS THAT ARE GAMES.**

**I have provided different templates for your consideration throughout the course.**

**If you design your own Unique Final Project remember that you must use a data source. This could be either a random- access file, MySQL file, MS Access Database, or XML File.**

**You may NOT use a TEXTFILE!!!!**

**Also, if you choose this option you must create a totally Unique Class Object.**

**(Not Emprec or Employee or Product !!!)**

**Please note that should you use a MySQL file you will receive significant extra credit.**

**You’ll see that there are many things for which you will receive extra credit should you choose to do so!**

1. **You must create an object and use the four major data types. These are the int, char, String and double. You may have more than 4 data fields should you like but make sure that you have shown me you can read and write each of the 4 data types. Use all the fields you need to.**
2. **You must use a toString().**
3. **In addition to the toString() you must have an additional 3 methods that process data.**

**These methods should also take in parameters and not be void methods!!!!!! In other words, your methods should return a value.**

**For Extra credit you may also**

**Inherit a class.**

**Implement an Interface.**

**Inherit an abstract class.**

**Use the Java Date Object to display the date in your output.**

**Use Constructors and not just a default constructor!**

**Use Java FX. If you write your Final in Java FX you will receive significant extra credit!**

**Use a TableView() lot’s of extra credit here!!!!**

**Develop one or more packages on your own. Packages designed by Netbeans, Eclipse etc. do not count as you don’t demonstrate that you learned how to do them.**

**Now Look at program Assignment21A.java found in the Java Files**

**Use a MessageDialogBox like that found on lines 170-176**

**Of Assignment21A.java**

**Use formatting of doubles and currency (money)**

**Please note that all these things that I require you do and all that are available for Extra Credit may be found in the one or more of the 3 Templates I provided!!!!**

**Let me know should you have any questions or concerns!**

**Good luck,**

**Jack**

**339 222 4247 (cell)**

Here is an example for what you might do for the Final Project!

**Emprec.java**

**/\*\***

**\* Employee Payroll class**

**\***

**\*/**

**//package pkg;**

**import java.text.DecimalFormat;**

**import java.util.Date;**

**public class Emprec extends HoursRate implements Emprec\_Iface{**

**//define properties**

**private int record;**

**private String jobTitle;**

**private String name;**

**private int age;**

**private int dependents;**

**private char gender;**

**private Date date;**

**//constructor**

**public Emprec( int record, String jobTitle, String name, int age, int dependents, char gender, double hours, double rate)**

**{**

**super(hours, rate);**

**this.record=record;**

**this.jobTitle=jobTitle;**

**this.name=name;**

**this.age=age;**

**this.dependents=dependents;**

**this.gender = gender;**

**date = new Date(); //initialize a date**

**}**

**public Date getDate(){**

**return date;**

**}**

**//interface method implementation**

**public double calc\_gross\_pay()**

**{**

**return ( getHours() \* getRate() );**

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

**//interface method implementation**

**public double calc\_fed\_tax(double hours,double rate)**

**{**

**double yearly\_income;**

**yearly\_income = hours \* rate \* TOTAL\_WEEKS; //52 - weeks per year**

**if (yearly\_income < YEAR\_INCOME\_S) return (hours \* rate \* TAX\_RATE\_ONE);**

**else if ( yearly\_income < YEAR\_INCOME\_M) return (hours \* rate\* TAX\_RATE\_TWO);**

**else return (hours \* rate \* TAX\_RATE\_THREE);**

**}// calc\_fed\_tax**

**//interface method implementation**

**public double calc\_state\_tax(double hours, double rate)**

**{**

**return (hours \* rate \* STATE\_TAX\_RATE);**

**}// calc\_state\_tax**

**public String toString()**

**{**

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

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

**return**

**("\n Record: "+ this.record +**

**"\n Job title: "+ this.jobTitle +**

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

**"\n Age: "+ this.age +**

**"\n Dependents: " + this.dependents +**

**"\n Gender: " + this.gender +**

**"\n Hours: " + twoDigits.format(getHours()) +**

**"\n Rate: " + dollar.format(getRate()) +**

**"\n Gross Pay: " + dollar.format(calc\_gross\_pay()) +**

**"\n Federal Tax: " + dollar.format(calc\_fed\_tax(getHours(), getRate())) +**

**"\n State Tax: " + dollar.format(calc\_state\_tax(getHours(), getRate())));**

**}//toString**

**}**

**Emprec\_Iface.java**

**/\*\***

**\* Midterm assignment - Employee Payroll class**

**\***

**\*/**

**//package pkg;**

**//create an interface**

**public interface Emprec\_Iface{**

**//define and initialize taxRate**

**public double TAX\_RATE\_ONE = 0.28;**

**public double TAX\_RATE\_TWO = 0.32;**

**public double TAX\_RATE\_THREE = 0.38;**

**public double YEAR\_INCOME\_S = 30000.00;**

**public double YEAR\_INCOME\_M = 50000.00;**

**public double STATE\_TAX\_RATE = 0.0561;**

**public int TOTAL\_WEEKS = 52;**

**public double calc\_gross\_pay();**

**public double calc\_fed\_tax(double hours,double rate);**

**public double calc\_state\_tax(double hours, double rate);**

**}**

**HoursRate.java**

**/\*\***

**Employee Payroll class**

**\*/**

**//package pkg;**

**//create class that hold data from hours and rate fields**

**public class HoursRate{**

**private double hours;**

**private double rate;**

**//default constructor**

**public HoursRate(){**

**this.hours = 0.00;**

**this.rate = 0.00;**

**}**

**//constructor**

**public HoursRate(double hours, double rate){**

**this.hours = hours;**

**this.rate = rate;**

**}**

**public HoursRate(String hours, String rate){**

**try{**

**this.hours = Double.valueOf(hours).doubleValue();**

**this.rate = Double.valueOf(rate).doubleValue();**

**}catch(NumberFormatException errmsg){**

**System.out.println("Invalid format " + errmsg);**

**this.hours = 0.0;**

**this.rate = 0.0;**

**}**

**}**

**//define accessor functions**

**public double getHours(){**

**return this.hours;**

**}**

**public double getRate(){**

**return this.rate;**

**}**

**}**

**MakePayrollFile.java**

**/\*\***

**\* Midterm assignment - Employee Payroll class**

**\*/**

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

**import java.awt.\*;**

**import java.awt.event.\*;**

**public class MakePayrollFile extends Frame //Frame class reperesents top-level winow with a title and border**

**implements ActionListener { //ActionListener interface - receive action events**

**// text components that allow for editing of single line of text**

**private TextField RecNumberField, NameField, DependentsField,**

**AgeField, HoursField,RateField, JobTitleField, GenderField;**

**private Button enter, // send record to file**

**done; // quit program**

**// Application other pieces**

**private DataOutputStream output;**

**public MakePayrollFile() //constructor**

**{**

**super( "Payroll File - Write" );**

**// Open the file**

**try {**

**output = new DataOutputStream(**

**new FileOutputStream( "payroll.txt" ) );**

**}**

**catch ( IOException e ) {**

**System.err.println( "File not opened properly\n" +**

**e.toString() );**

**System.exit( 1 );**

**}**

**setSize( 400, 400 ); // set frame size**

**setLayout( new GridLayout( 9, 2 ) ); // create layout with 7 rows and 2 columns**

**// create the components of the Frame**

**add( new Label( " Record Number" ) );**

**RecNumberField = new TextField();**

**add( RecNumberField );**

**add( new Label( " Job Title" ) ); //add new text field**

**JobTitleField = new TextField();**

**add( JobTitleField );**

**add( new Label( " Name" ) );**

**NameField = new TextField( );**

**add( NameField );**

**add( new Label( " Age" ) );**

**AgeField = new TextField( );**

**add( AgeField );**

**add(new Label(" Dependents "));**

**DependentsField = new TextField();**

**add(DependentsField);**

**add(new Label(" Gender "));**

**GenderField = new TextField();**

**add(GenderField);**

**add( new Label( " Hours" ) );**

**HoursField = new TextField( );**

**add( HoursField );**

**add( new Label( " Rate" ) );**

**RateField = new TextField( );**

**add( RateField );**

**enter = new Button( " Enter" );**

**enter.addActionListener( this );**

**add( enter );**

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

**done.addActionListener( this );**

**add( done );**

**setVisible( true ); //set visability of frame**

**HoursField.requestFocus(); //set mouse focus in a field Hours**

**}**

**public void actionPerformed( ActionEvent e ) //get mouse events**

**{**

**if ( e.getSource() == enter ) addRecord(); //add new record**

**if ( e.getSource() == done ) { //close a file**

**try {**

**output.close();**

**}**

**catch ( IOException io ) {**

**System.err.println( "File not closed properly\n" + io.toString() );**

**System.exit( 1 );**

**}**

**System.exit( 0 );**

**}**

**}**

**public void addRecord(){**

**//define temporary variables**

**int recNumber = 0;**

**int ageTemp = 0;**

**int depenTemp = 0;**

**Double h,r;**

**char genderTemp;**

**if(!RecNumberField.getText().equals( "" )) { //check if input is not empty**

**try{**

**recNumber = Integer.parseInt( RecNumberField.getText());**

**if ( recNumber > 0 ) { //if account number is more than zero**

**output.writeInt( recNumber ); //write integer RecNumber to a file**

**output.writeUTF( JobTitleField.getText() );**

**output.writeUTF( NameField.getText() );**

**try{**

**ageTemp = Integer.parseInt(AgeField.getText()); //convert str to int**

**output.writeInt(ageTemp); //write age**

**}catch(NumberFormatException nfe){**

**System.err.println("You must enter an integer number" );**

**}**

**try{**

**depenTemp = Integer.parseInt(DependentsField.getText());**

**output.writeInt(depenTemp);**

**}catch(NumberFormatException nfe){**

**System.err.println("You must enter an integer number" );**

**}**

**try{**

**genderTemp = GenderField.getText().charAt(0);**

**output.writeChar(genderTemp);**

**}catch(IndexOutOfBoundsException nfe){**

**System.err.println( "Gender must be entered as a one character" );**

**}**

**try{**

**h = new Double ( HoursField.getText() );**

**output.writeDouble( h.doubleValue() );**

**}catch(NumberFormatException nfe){**

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

**}**

**try{**

**r = new Double ( RateField.getText() );**

**output.writeDouble( r.doubleValue() );**

**}catch(NumberFormatException nfe){**

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

**}**

**}//if statement**

**// clear the TextFields**

**RecNumberField.setText( "" );**

**JobTitleField.setText("");**

**NameField.setText( "" );**

**AgeField.setText("");**

**GenderField.setText("");**

**DependentsField.setText("");**

**HoursField.setText( "" );**

**RateField.setText("");**

**}catch(NumberFormatException nfe){**

**System.err.println("You must enter an integer number" );**

**}catch ( IOException io ) {**

**System.err.println("Error during write to file\n" + io.toString() );**

**System.exit( 1 );**

**}**

**}//if**

**}//addRecord**

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

**{**

**new MakePayrollFile();**

**}**

**}**

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

**ReadPayrollFile.java**

**/\*\***

**\* Midterm assignment - Employee Payroll class**

**\***

**\*/**

**// Major Assignment !!!!!!!!**

**// In the calc\_fed\_tax method create an interface to set the**

**//tax rates as constants.**

**// Also, in the interface, declare the**

**// double calc\_gross\_pay()**

**// and the**

**// double calc\_fed\_tax(double hours,double rate)**

**//methods. These are then implememnted by the class Emprec.**

**// Add textfields and labels to display the gross pay, fed tax and state tax.**

**// Create an class that the class Emprec inherits to bring in the hours and rate fields.**

**// remember that you must then remove them from the Emprec class.**

**// This program reads a file sequentially and displays each**

**// record in text fields.**

**//import pkg.Emprec;**

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

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.text.DecimalFormat;**

**import java.text.SimpleDateFormat;**

**import java.util.Date;**

**public class ReadPayrollFile extends Frame**

**implements ActionListener {**

**Emprec employee;**

**private TextField RecNumberField, NameField, DependentsField, AgeField, HoursField,**

**RateField, JobTitleField, GenderField, GrossPayField, FedTaxField, StateTaxField, DateField;**

**private Button next, // get next record in file**

**done; // quit program**

**// Application other pieces**

**private DataInputStream input;**

**// Constructor -- initialize the Frame**

**public ReadPayrollFile()**

**{**

**super( "Read Client File" );**

**// Open the file**

**try {**

**//open a file**

**input = new DataInputStream(**

**new FileInputStream( "payroll.txt" ) );**

**}**

**catch ( IOException e ) {**

**System.err.println( "File not opened properly\n" +**

**e.toString() );**

**System.exit( 1 );**

**}**

**setSize( 400, 500 );**

**setLayout( new GridLayout( 13, 2 ) );**

**// add Text Filds**

**add( new Label(" Date"));**

**DateField = new TextField();**

**DateField.setEditable(false);**

**add(DateField);**

**add( new Label( " Record Number" ) );**

**RecNumberField = new TextField();**

**RecNumberField.setEditable( false );**

**add( RecNumberField );**

**add( new Label( " Job Title" ) );**

**JobTitleField = new TextField();**

**JobTitleField.setEditable(false);**

**add( JobTitleField );**

**add( new Label( " Name" ) );**

**NameField = new TextField();**

**NameField.setEditable( false );**

**add( NameField );**

**add( new Label( " Age" ) );**

**AgeField = new TextField( );**

**AgeField.setEditable(false);**

**add( AgeField );**

**add(new Label(" Dependents "));**

**DependentsField = new TextField();**

**DependentsField.setEditable(false);**

**add(DependentsField);**

**add(new Label(" Gender "));**

**GenderField = new TextField();**

**GenderField.setEditable(false);**

**add(GenderField);**

**add( new Label( "Hours" ) );**

**HoursField = new TextField();**

**HoursField.setEditable( false );**

**add( HoursField );**

**add( new Label( "Rate" ) );**

**RateField = new TextField();**

**RateField.setEditable( false );**

**add( RateField );**

**//add new text fields**

**add( new Label("Gross Pay") );**

**GrossPayField = new TextField();**

**GrossPayField.setEditable( false );**

**add(GrossPayField);**

**add( new Label( "Federal Tax" ) );**

**FedTaxField = new TextField();**

**FedTaxField.setEditable( false );**

**add(FedTaxField);**

**add( new Label("State Tax") );**

**StateTaxField = new TextField();**

**StateTaxField.setEditable( false );**

**add(StateTaxField);**

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

**next.addActionListener( this );**

**add( next );**

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

**done.addActionListener( this );**

**add( done );**

**setVisible( true );**

**}**

**public void actionPerformed( ActionEvent e )**

**{**

**if ( e.getSource() == next )**

**readRecord();**

**else**

**closeFile();**

**}**

**public void readRecord() //read record from a file**

**{**

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

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

**//define temporary variables**

**int recnumber = 0;**

**int age = 0;**

**int depTemp = 0;**

**String name, jobTitle;**

**double h,r;**

**char gender;**

**// input the values from the file**

**try {**

**//read data from a file**

**recnumber = input.readInt();**

**jobTitle = input.readUTF(); //read one character**

**name = input.readUTF();**

**age = input.readInt();**

**depTemp = input.readInt();**

**gender = input.readChar();**

**h = input.readDouble();**

**r=input.readDouble();**

**//create new instance of Emprec object**

**employee=new Emprec(recnumber, jobTitle, name, age, depTemp, gender, h, r);**

**System.out.println(employee); //toString()**

**//format date and time**

**SimpleDateFormat dateFormat = new SimpleDateFormat("MM/dd/yyyy, hh:mm");**

**Date date = employee.getDate();**

**//show data on fields**

**DateField.setText(String.valueOf(dateFormat.format(date)));**

**RecNumberField.setText( String.valueOf( recnumber ) );**

**JobTitleField.setText(jobTitle);**

**NameField.setText( name );**

**AgeField.setText(String.valueOf(age));**

**DependentsField.setText(String.valueOf(depTemp));**

**GenderField.setText(String.valueOf(gender));**

**HoursField.setText( String.valueOf( twoDigits.format(h) ) );**

**RateField.setText( String.valueOf( dollar.format(r) ) );**

**GrossPayField.setText( String.valueOf( dollar.format(employee.calc\_gross\_pay())));**

**FedTaxField.setText( String.valueOf( dollar.format(employee.calc\_fed\_tax(h, r))));**

**StateTaxField.setText( String.valueOf( dollar.format(employee.calc\_state\_tax(h, r))));**

**}**

**catch ( EOFException eof ) { //end of file exception**

**closeFile();**

**}**

**catch ( IOException e ) {**

**System.err.println( "Error during read from file\n" + e.toString() );**

**System.exit( 1 );**

**}**

**}//end readRecord()**

**private void closeFile()**

**{**

**try {**

**input.close();**

**System.exit( 0 );**

**}**

**catch ( IOException e ) {**

**System.err.println( "Error closing file\n" + e.toString() );**

**System.exit( 1 );**

**}**

**}**

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

**{**

**new ReadPayrollFile();**

**}**

**}**

**Assignment21.java**

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

**import java.text.\*;**

**import javax.swing.JOptionPane;**

**//import java.swing.\*;**

**class Emprec {**

**String name;**

**String address;**

**String socsec;**

**double hours;**

**double rate;**

**double dept\_credit;**

**char sex;**

**int age;**

**boolean active;**

**Emprec (String name,String address,String socsec,String hours,String rate,**

**String dept\_credit,String sex,String age,String active)**

**{**

**try**

**{**

**this.name=name;**

**this.address=address;**

**this.socsec=socsec;**

**this.hours=Double.valueOf(hours).doubleValue();**

**this.rate=Double.valueOf(rate).doubleValue();**

**this.dept\_credit=Double.valueOf(dept\_credit).doubleValue();**

**this.sex=sex.charAt(0);**

**this.age=Integer.parseInt(age);**

**this.active=Boolean.valueOf(active).booleanValue();**

**}**

**catch(NumberFormatException errmsg)**

**{**

**System.out.println("Invalid format"+ errmsg);**

**this.name = "";**

**this.hours = 0.0;**

**this.rate = 0.0;**

**this.dept\_credit=0.0;**

**this.age=0;**

**this.active=false;**

**}//catch**

**}//Emprec constructor !!!!**

**double calc\_fed\_tax(double hours,double rate)**

**{**

**double yearly\_income;**

**yearly\_income = hours \* rate \*52;**

**if (yearly\_income < 30000.00) return (hours \* rate \*.28);**

**else if ( yearly\_income < 50000.00 ) return (hours \* rate\* .32);**

**else return (hours \* rate \* .38);**

**}// calc\_fed\_tax**

**double calc\_gross\_pay()**

**{**

**return (hours \* rate);**

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

**double calc\_state\_tax(double hours, double rate)**

**{**

**double state\_tax;**

**state\_tax = hours \* rate \* .0561;**

**return (state\_tax);**

**}// calc\_state\_tax**

**double calc\_netpay(double hours,double rate,double dept\_credit)**

**{**

**double netpay;**

**// double grosspay=calc\_gross\_pay();**

**//double sTtax=calc\_state\_tax(hours, rate);**

**//double fedtax=calc\_fed\_tax(hours, rate);**

**return (calc\_gross\_pay()- calc\_state\_tax(hours, rate) -calc\_fed\_tax(hours, rate) - dept\_credit);**

**}// calc\_netPay**

**public String toString()**

**{**

**NumberFormat fmt = NumberFormat.getCurrencyInstance();**

**return**

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

**"\n Address: \t\t"+ address +**

**"\n Social Security Number: "+ socsec +**

**"\n Hours: \t\t\t"+ hours +**

**"\n Rate: \t\t\t"+ fmt.format(rate) +**

**"\n Sex: \t\t\t"+ sex +**

**"\n Age: \t\t\t"+ age +**

**"\n Status: \t\t" + (this.active ? "Active" : "Inactive") +**

**"\n Gross pay: \t\t"+ fmt.format(calc\_gross\_pay()) +**

**"\n State tax: \t\t"+ fmt.format(this.calc\_state\_tax(this.hours,this.rate))**

**+**

**"\n Net Pay : \t\t"+ fmt.format(this.calc\_netpay(this.hours,this.rate,this.dept\_credit))**

**);**

**}//toString**

**}// Emprec**

**class Assignment21**

**{**

**public static void main(String args[]) throws IOException**

**{**

**BufferedReader inData = new BufferedReader(new InputStreamReader(System.in));**

**// create strings for the input data for the Emprec object**

**String str\_name;**

**String str\_address;**

**String str\_socsec;**

**String str\_hours;**

**String str\_rate;**

**String str\_sex;**

**String str\_age;**

**String str\_dept\_credit;**

**String str\_active;**

**NumberFormat fmt = NumberFormat.getCurrencyInstance();**

**for(;;){**

**//READ THE FILE !!!!**

**str\_name = inData.readLine();**

**if(str\_name.equalsIgnoreCase("exit")) System.exit(0);**

**str\_address = inData.readLine();**

**str\_socsec = inData.readLine();**

**str\_hours = inData.readLine();**

**str\_rate = inData.readLine();**

**str\_sex = inData.readLine();**

**str\_age = inData.readLine();**

**str\_dept\_credit = inData.readLine();**

**str\_active= inData.readLine();**

**Emprec employee=new Emprec(str\_name,str\_address,str\_socsec,str\_hours,str\_rate,str\_dept\_credit,**

**str\_sex,str\_age,str\_active);**

**System.out.println(employee);**

**System.out.println(" Federal tax:\t\t"**

**+ fmt.format(employee.calc\_fed\_tax(employee.hours, employee.rate)));**

**System.out.println(" Net Pay :\t\t" +**

**fmt.format(employee.calc\_netpay(employee.hours, employee.rate, employee.dept\_credit)));**

**JOptionPane.showMessageDialog(null,**

**"Employee Information:" +**

**employee +**

**"\n Net Pay: " + fmt.format(employee.calc\_netpay(employee.hours,**

**employee.rate, employee.dept\_credit)),**

**"Employee Payroll Being Processed",**

**JOptionPane.PLAIN\_MESSAGE );**

**}//for**

**}//main**

**}//Assignment21**

**dựa trên đề bài này , tôi có những yêu cầu sau :**

1. **chủ đề: quản lí kho ô tô**
2. **tạo 1 package chứa 4 class :**

* **abstract class vehicle**

**( string id ,**

**string name ,**

**int number ,**

**char filter )**

**Abstract method toString**

* **interface currency**

**( double price )**

**double change( string from , string to )**

* **class car extend vehicle implement currency**

**( date ngày nhập kho ,**

**char type )   
các method set , get , bool check\_exist( string name )**

* **class view ( hiển thị list car trong table view , dùng java fx )**
* **class model ( có các hàm , add 1 car , get all car , delete 1 car )**
* **class controller**

**( biến list car , biến model )**

**Method list<car> getwithfilter( char \_fiter )**