Introduction to Computer Science II
Assignment 4

Due: 4:00 p.m. June 29, 2016

Programming Graphical User Interfaces in Java: Living Expenses Calculator

You are to create a Living Expenses Calculator application (not an applet) with a GUI, using the Java language and Swing libraries. The application helps the user tracks his or her expenses per year, by tracking the various monthly expenses for different categories and by taking into consideration the user's yearly income and tax. The user of your program should be able to enter the following:

- Housing expenses (rental or mortgage, insurance amount, and bills for utilities)
- Transportation expenses (fuel, transportation tickets or subscription, and parking)
- Living essentials expenses (groceries, clothing, personal care, and medication)
- Life style expenses (dinning out and vacation)
- Income before tax (the gross amount earned per year)
- Tax amount (a percentage applied to gross income as deduction, e.g., 30%)

Assume that payment expenses are made monthly.

An example of the expenses would be:

- Housing expenses (rental or mortgage = $1300, insurance amount = $200, and bills for utilities = $250)
- Transportation expenses (fuel = $50, transportation tickets or subscription = $100, and parking = $50)
- Living essentials expenses (groceries = $400, clothing = $50, personal care = $50, and medication = $100)
- Life style expenses (dinning out = $120 and vacation = $40)
- Income before tax = $60000
- Tax amount = 0.3

Your program will calculate and display the following:

- The net income (the income after tax): yearly and monthly. Divide by 12 to get the monthly net income
- The total expenses of a particular month (e.g., the total of housing, transportation, living-essentials, and life style expenses)
- The month with the highest and lowest expenses
- Any savings at the end of the year
- The total expenses of a particular category (e.g., transportation essentials) for a particular month or for the whole year.

Use whatever GUI widgets you feel are appropriate. Follow the Model-View-Controller design pattern. Put all your classes into a package called lab4.

Bonus

- Allow the system to handle unexpected/invalid input. (2 points)
- Allow the user to create a monthly budget for his or her expenses and payments, and inform when it goes over/under the budget. (4 points)

- Allow the user to edit and customize the categories as needed such as removing a sub-item from the transportation expenses category (4 points)

Documentation

Create a UML class diagram that documents all of your classes and their relationships (do not show any of the classes from the Java libraries). Hand-drawn notation is acceptable, although you can use a graphics program if you wish.

Use javadoc to create HTML documentation for all your classes (but ignore classes from the Java libraries). Be sure all the classes and all the methods are documented; they must be declared public for javadoc to work.

New Skills Needed for this Assignment:

- Understanding of event-driven programming
- Use of the JDK event model
- Use of AWT and Swing GUI elements in the Java API, including Components, Containers, and Layout Managers
- Understanding of the Model-View-Controller design pattern
- Understanding and use of Exceptions
- Understanding and use of interfaces
- Understanding and use of packages

Submit the following:

1. Your Java source code via electronic submission. Use the *Assignment 4* Dropbox Folder in D2L to submit electronically. Your TA will compile and run your program to test it. Your TA may also have you demonstrate your application in a tutorial session.
2. A *Readme.txt* file, which explains how to compile your source code and how to run the program. Submit this electronically to the D2L dropbox.
3. Your HTML documentation produced by running *javadoc* on your source code. Also submit this to the D2L drop box. Your TA will check the HTML documentation to make sure it is complete and correct. Your TA may also have you show this when you demonstrate your application in a tutorial session.
4. Your class diagram using UML notation. Submit this electronically to the D2L drop box. The file should be in a commonly used file format such as PDF, TIFF, or JPEG.

Introduction to Computer Science II
Assignment 4 Grading

Student: ____

Graphical User Interface			
Input elements	14	______	
Output elements	16	______	
Design quality and ease of use	5	______	
(e.g., good layout, coloring, simplicity) https://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int_design.html			
Class Structure and Quality			
Use of the MVC design pattern	5	______	
Miscellaneous			
Code structure (documentation, formatting, etc.)	4	______	
Detailed UML Class Diagram	4	______	
HTML Documentation	2	______	
Total	**50**	______	______%
Bonus			
Handle Unexpected issues	2	______	
Custom Expenses Categorization	4	______	
User Budget	4	______	
Total Bonus (10% for 10/10)	**10**	______	______%
Assignment Grade	**60**	______	______%