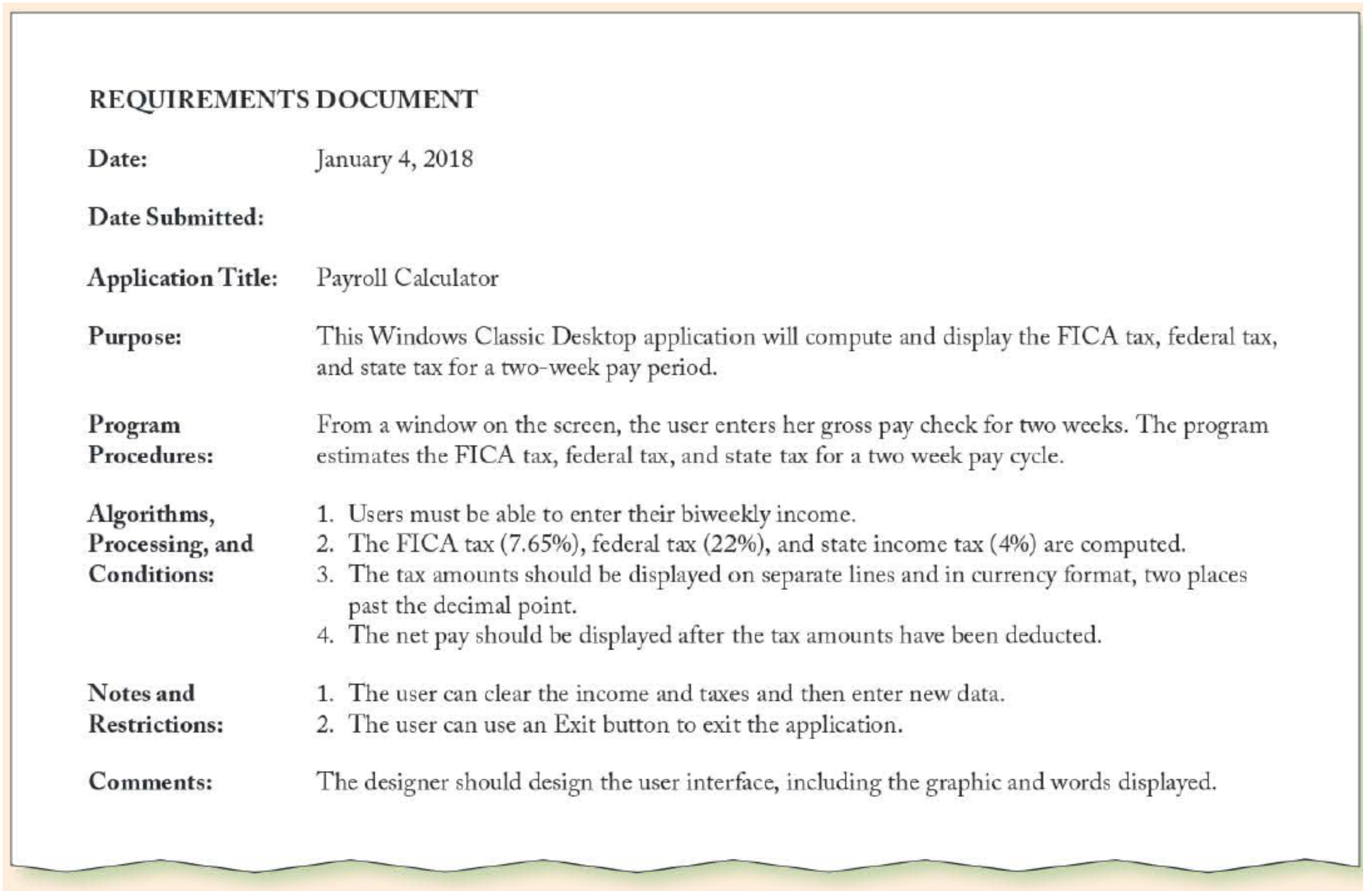
Assignment P2: Payroll Calculator GUI Development Project – 75 Points for this assignment (possible 10 point bonus)

Due: Tuesday 2/27 by midnight

Project Overview

This assignment will be the creation of an event planning document, a standalone functional GUI-based app, and a README markdown file. (The assignment is taken from Hoisington Chapter 4 pg. 224, but you do not need the book for the work.)

You may choose whether to develop your application in VS/VB or in Python/Tkinter. Tkinter users can optionally use PAGE to help with layout/development. The requirements for your application are as follows:

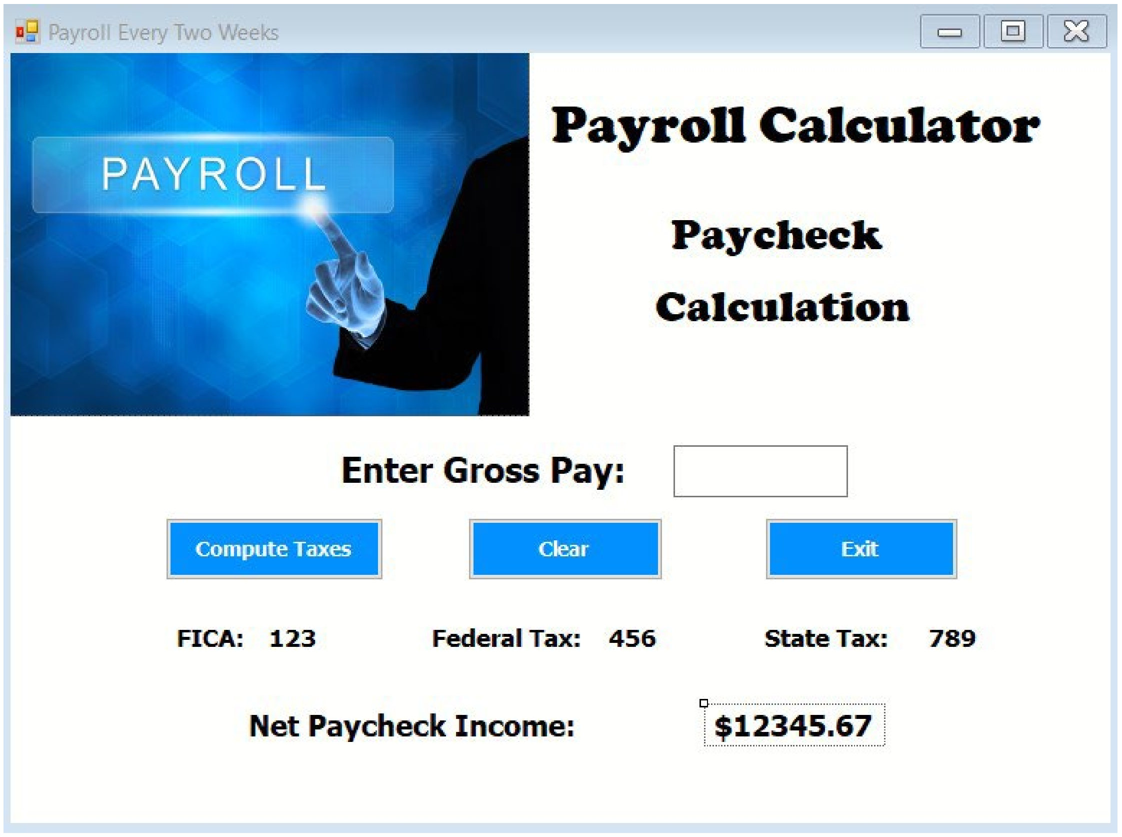


REQUIREMENTS DOCUMENT

Date:	January 4, 2018
Date Submitted:	
Application Title:	Payroll Calculator
Purpose:	This Windows Classic Desktop application will compute and display the FICA tax, federal tax, and state tax for a two-week pay period.
Program Procedures:	From a window on the screen, the user enters her gross pay check for two weeks. The program estimates the FICA tax, federal tax, and state tax for a two week pay cycle.
Algorithms, Processing, and Conditions:	1. Users must be able to enter their biweekly income. 2. The FICA tax (7.65%), federal tax (22%), and state income tax (4%) are computed. 3. The tax amounts should be displayed on separate lines and in currency format, two places past the decimal point. 4. The net pay should be displayed after the tax amounts have been deducted.
Notes and Restrictions:	1. The user can clear the income and taxes and then enter new data. 2. The user can use an Exit button to exit the application.
Comments:	The designer should design the user interface, including the graphic and words displayed.

FIGURE 4-98

A mockup of the application has been provided (along with the payroll JPG image):



First Deliverable: Create an Event Planning Document – it should include what will need to happen for each interaction with the interface per the requirements provided. Use the following template (or something like it) for your Event Plan:

Object	**Event Trigger**	**Event Processing**

Second Deliverable: Create a working GUI that performs the computations as laid out in the requirements document and updates the form accordingly.

Third Deliverable: Create a README Markdown file for Assignment P1, including Project name and student name, development tools used, and any issues you encountered in development (none is a possible answer).

GUI Development Details

Use naming conventions for controls and variables appropriate to VB or Python/Tkinter.

When the form is loaded (VB Form Load event) OR when the Clear button is clicked:
1. The example text is removed from the four label controls (FICA, Federal Tax, State Tax, Net Paycheck Income).
2. The Gross Pay income textbox is empty and has focus.

Exit button click event:
1. Program closes.

Compute Taxes click event (all variables are shown in VB format, Python coders can use these forms or use similar naming):
1. Write code to create the following variables (presented in VB format):
```
strIncome, decIncome, decFica, decFederal, decState, decNet
```

2. Write code to set the following constants and initialize them to the values specified in the requirements:
```
cdecFica 0.0765
cdecFed 0.22
cdecState 0.04
```

3. Convert the string strIncome value from the income textbox to the floating point decIncome.

4. Calculate the three tax amounts as a percentage of decIncome (hint: simply multiply)

5. Calculate the net income by subtracting the three tax amounts from decIncome (hint: there will be something left over / the result will be a positive number.)

6. Display the results on the form. Where appropriate, convert numbers to currency ($ and two decimal places)

7. Convert the three tax amounts and the net income amount to String and assign each to the Text property of the matching label control.

Possible 10 point bonus: Input Validation and Message Box use

Have your code check the input strIncome sting to see if it is a valid number before converting it to a float or decimal value. If it is not a valid number, display a message box alerting the user that the entry is not valid. When the message box has been closed, clear the Gross Pay income textbox and give it focus for reentry.

It is suggested that you do not add the bonus functionality until all other functions above are working as you expect.

Project Delivery and Rubric

Submit your VB project files or Python files for your application, along with your Event Planning document, and your README in a GitHub repo. Provide the GitHub repo link as a comment or as content in a text file.

Project grading rubric:

- 5 points – README as requested above
- 20 points – A thorough Event Plan document identifying actions needed and controls involved for all possible program events
- 20 points – All objects requested represented in code and visually on GUI
- 20 points – GUI should run as requested in development environment (VB or Python) – all events should occur as requested
- 10 points – Cleanly formatted code with comments. This should include a header block containing student, project, and class names. Comments for functions/methods/classes, comments for key actions or any statements that may not be obvious in their function. Use descriptive variable names.
- 10 point bonus – added input validation and message box use – this bonus is all or nothing, both input validation and message box behaviors must work to be awarded.

Note: Always cite what you write! If you get code or content from somewhere you must include at least a URL or other source identification. You must understand all the code you turn in. It is plagiarism (academic dishonesty) to use code or content, in part or in whole, written by other people without proper attribution. Failure to do so will result in a 0 on the assignment and may result in an academic misconduct report.

See Bruce for questions.