

ISM 3232 – Travis – Individual Technical Assignment #2

ASSIGNMENT OBJECTIVES

- To repeat the Objectives of Individual Assignment #1:
 - Use Visual C# 2015/2017/2019 to create a project. (***) See the next page for the correct project name to use. (***)
 - Observe professional programming style guidelines (e.g., comments, indentation, etc.).
 - Determine the most appropriate C# controls to use to accomplish the requirements of the program.
 - Properly declare all necessary constants and variables with appropriate data types and scopes.
 - Follow appropriate standardized naming conventions for constants, variables, and controls.
 - Write code to accept input from the user, perform calculations, and convey output to the user.
 - Format output so that it displays appropriately for the situation involved (e.g., as currency).
- Utilize appropriate decision structures to branch code execution as required by the program.
- Share event handler code written for one control with other controls.
- Create a custom message box that solicits input from the user and then responds appropriately to the input.
- Take data the user has input into controls on the form and summarize it in a custom message box.

BUSINESS SCENARIO

Language Arts Institute (LAI) is an online language training software business that offers self-paced training software to students enrolled at a local community college who are seeking a foreign language literacy certification. *LAI* only supplies the training software; the college handles the testing and certificate granting. But, *LAI* must follow certain rules (e.g., quantity limitations per student per school term) established by the college regarding the sale of the software. Orders from students are taken by phone or in person by clerks at a sales desk in the student union. Currently, the company clerks write the orders down on paper forms. However, the company wishes to automate its order-taking function with a Windows application running on laptop computers. The first step in this process is to develop a prototype of a simple order entry form as described below.

ASSIGNMENT REQUIREMENTS

- There is to be a picture box control placed at the top of the form that displays a small graphic file of your choosing related to languages or training. Then, four group boxes should be used, as described below.
- Term Information – LAI must keep track of the school term in which each purchase is made. This information consists of both a term specification (Fall or Spring only) and a year specification (e.g., 2018). Utilize radio buttons for the term (with Fall being the default), and utilize a combo box for the year, with the list including the years 2019 through 2024. The combo box should not allow any other year entry by the user. When the program begins, the combo box should not display any year until the user selects one from the list.
- Student Information – The Student ID, first name, last name, and email address are to be entered for each student. (The Student ID is actually the student's Social Security Number and should follow the standard format for that number.) Also, the student's residence status must be recorded, with options of In-State and Out-Of-State. (The software is priced lower for in-state students – see below). In-State should be the default setting.
- Course Order Information – The software packages are referred to as “courses.” The offerings currently include five courses: Beginning French, Beginning German, Beginning Italian, Beginning Russian, and Beginning Spanish. The charge for the software is \$59 per course for in-state students and \$109 per course for out-of-state students. There is no sales tax charge for these purchases. The maximum number of courses that a student may purchase in a single order is three. (There are no other restrictions placed on the sale of courses.)
- Payment Information – Only MasterCard and Visa credit cards are accepted as payment. Use radio buttons for these options. MasterCard should be set as the default selection for credit card type. The 16-digit credit card number and the card expiration date must also be recorded. Use a 00/00/0000 format for the expiration date.
- There are to be three buttons in a row at the bottom of the form labeled: Save, Clear, and Exit (from left to right). There should be an Access Key defined for each of these buttons. Also, there should be an appropriate ToolTip created for each of the buttons. (For this assignment, no other form controls require Access Keys or ToolTips.)
- As the user clicks on courses, the program is to automatically update two fields on your form: the total number of courses purchased on the order and the total price of the order (which will be calculated as the number of courses times the price per course for that student). (*Hint: Create a check box control for each course and use the CheckedChanged event handler for the check boxes to run the necessary shared event handler code.*) At program startup, these two fields (total number of courses purchased and total price of the order) should display initial values of 0 and \$0.00, respectively. The form should also display the current price per course.

- When the Save button is clicked, the program is not to actually save any data to an external file. Instead, the saving of data will be simulated by displaying the entire contents of the order in a message box. This message box should have appropriate text in its title bar and should include an Information icon. The message box should display the following data, properly labeled, with one data item per line: registration term (term and year), student ID, student name (first and last), email address, residence status, total courses purchased, price per course, total order price, credit card type, card number, and card expiration date.
EXTRA CREDIT: Also display, after the previous data, the name of each course ordered (one course per line).
- Do not “save” an order for less than one or more than three courses. When the Save button is clicked, if there is no course selected, or more than three courses selected, instead of displaying the message box described above, a message box should display stating that a course order must contain at least one course but no more than three courses. This message box should have appropriate text in its title bar and should include an Exclamation icon.
- Clicking the Clear button is to return the form to its original state, clearing and resetting all data entry areas and calculated fields, and then sending the focus to the first data entry control on the form. (*Hint: A combo box control can have its text area cleared by setting its SelectedIndex property to a value of -1.*)
- Clicking the Exit button is to terminate the program using the Close method. However, when the button is clicked, the program must first prompt the user with a message asking “Are you sure you wish to quit?” The message box is to provide response options of Yes and No. The message box should also have appropriate text in its title bar and should include a Question icon. If the user chooses the Yes button in the message box, the program should close. But, if the user chooses the No button, the program should not close.
- You are to give meaningful names to all controls used on the form, and all constants and variables used in your code. Follow standard C# naming conventions, as described in your textbook.
- Use appropriate labels for your controls to clearly identify them, and use appropriate text in the form’s title bar.
- Use group boxes, as described earlier, to appropriately group the form controls into logical sections.
- The alignment, spacing, and sizing of all controls should be appropriate, neat, and professional in appearance.
- When the program runs, the form should display in the center of the screen.
- The tab order should be set correctly so that focus flows logically through the controls on the form.
- Data that represents currency values should be displayed with the correct currency format.
- Data representing numeric values, including currency values, should be displayed right-aligned in controls, while text values (such as names) should be displayed left-aligned in controls.
- Use comments very liberally throughout your code, for each event handler and each significant block of code.
- Include an initial comment in your code, before all other code, similar to the following:

```
// Programmer: John Doe
// Project: YourLastName_2
// Description: Travis Individual Assignment #2
```

(Use your own name and project name, of course.)

DELIVERABLES

- Submit your assignment via Canvas following the instructions provided below. **This assignment due date is posted on Canvas.** *No late work will be accepted.*
- Name your C# project as: ***YourLastName_2*** (for example, Smith_2, if your last name is Smith). 
- If your program will not run, place a comment to that effect at the very top of your code. (However, it is better if you can get your program to run, even if it is incomplete. If necessary, comment out any code lines that prevent your program from running. Use comments to identify code lines that you have identified as being incorrect.)
Note: Any comments on the assignment should be made in your code, **not** on the Canvas submission page.)
- Use either Windows native compression utility or an archiving program such as 7-Zip or WinZip to zip your entire project folder into a single .zip file. Make sure that all folders and files that make up your project are included. (Refer to the handout titled “Zipping and Unzipping a Visual C# Project” for specific details.)
- Submit your zipped file via the class Canvas carefully following the same steps that you followed for Individual Assignment #1. (See the handout for that assignment for those instructions.)

This is an individual assignment. You are NOT to discuss it with other students or collaborate with other students in any way. If you have questions about this assignment, contact your instructor.