

**CALIFORNIA STATE UNIVERSITY, SACRAMENTO**  
College of Business Administration

MIS 15 – Introduction to Business Programming

Programming Assignment 3 (P3)

Points: 50

Due Date: Tuesday, May 10

The purpose of this assignment is to familiarize you with the basic concepts of object-oriented programming (OOP). In the past, you have worked with VB's GUI (graphical user interface) objects, such as labels, text, list and combo boxes. As you recall, they (objects) belong to a class from which they inherit their properties and methods. For example, all combo boxes have common properties that can be set to different values and methods that allow them to behave the same. The same applies for OOP, except you code the properties and methods.

This assignment involves building a base (or super) class and 3 derived (or sub) form classes, and a base class and 2 derived customer classes. The derived form classes will inherit most of their properties (i.e., form color, graphic, etc.) and methods (i.e., time, date, main menu with file and help menus) from the base class as noted in the figures on the last 2 pages of this assignment. The derived class forms will allow the program to interact with the user for different functions: menu, check overdraft recommender and new customer data entry.

Two customer type classes will inherit their properties and methods from a base class. Essentially, they are patterned from the base class. Instantiate all objects for regular (banking) customers in a separate class from objects of student customers.

## **Process**

### Menu Form

The program starts from the main menu (Figure 1). Currently, there are 2 options: check overdraft advisor and new customers. The user will click on the hypertext link for his/her selection and control is directed to the corresponding form. After the user has completed his/her selection, program control will return to this form. The termination of the program will also occur through this form (file menu → exit).

### Check Overdraft Advisor

The user enters the customer's name, account number, amount of the check and the current balance in the checking account, and indicates the account type (regular customer vs. student customer) and credit rating (good, poor and unknown) through radio buttons (Figure 2).

After all entries have been made, the user clicks on the “advise” button (with a flat appearance property). In advising the user, the program will evaluate the following five business rules:

Rule 1	If a student account has been indicated, do not pay the overdraft (i.e., not an account feature) Recommendation message: Do not pay overdraft; student account.
Rule 2	If the customer’s credit rating is poor, do not pay the overdraft. Recommendation message: Do not pay overdraft; credit rating poor.
Rule 3	If the customer’s credit rating is unknown AND the check exceeds (i.e., greater than) 10 percent of the balance, do not pay the overdraft. (This does not apply to student accounts.) Recommendation message: Do not pay overdraft; unknown credit rating and check exceeds 10 percent of the current balance.
Rule 4	If the customer’s credit rating is good and the check exceeds 50 percent of the balance, do not pay the overdraft. (This does not apply to student accounts.) Recommendation message: Do not pay overdraft; check exceeds 50 percent of the current balance.
Rule 5	If the customer’s credit rating is good and the check does not exceed or is equal to 50 of the balance, pay the overdraft. Recommendation message: Pay overdraft;

The user can pay or deny the overdraft. (For now, we will do nothing with these two buttons.)

The “cancel and clear” button clears all entries, including the text boxes, radio buttons and labels. The exit the form, the user clicks on the file menu and selects exit. Control returns to the menu form.

### New Customer

New customer objects are instantiated for the regular and student classes in the new customer form (Figure 3). The user enters the customer’s 8-digit account number, first and last names, and middle initial, the account type (regular or student), and the opening balance. Validate all entries as follows:

- Account number: 8 digits
- First and last names: presence
- Account type: one must be chosen
- Opening balance: numeric

If a violation occurs when the user clicks on the “add customer” button, a descriptive message box will appear and the focus placed on the respective text box.

The “clear” button removes all previous entries from the text boxes and unselects the radio buttons.

Depending on the account type (regular or student), the object will be instantiated in its corresponding class.

### Form Class

Create a base form class that all forms will inherit. The base will include a main menu with the file and help menus, the continuous time and day, and a graphic. The file menu contains an exit item (me.close) and the help menu an about item (message box with a description of software version).

Select a graphic and place it in a graphic box on the form.

The current time and day code is written in the load subprocedure as follows:

```
Private Sub baseForm_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    dateLabel.Text = Now.ToString("MMMM d, yyyy")
    timeLabel.Text = Now.ToLongTimeString
End Sub
```

*Where:*      dateLable and timeLabel              Labels on the form

Refer to the VB.net help function for descriptions on Now and ToLongTimeString. To keep the time current, create a subprocedure for the timer (drag the timer from the tool box to form and double click on the timer1 icon). Include the following code in the subprocedure:

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    timeLabel.Text = Now.ToLongTimeString
End Sub
```

Note. Be sure to enable the timer through its properties.

The three other derived form classes will inherit these properties.

### Customer, Regular and Student Classes

Create a customer class with its account number, customer name (first, last and middle initial), account type, opening balance properties, and current date and time of the transaction. Be sure to include the accessor methods and constructor. Instantiate a regular customer or student customer object based on the account type selection.

- Account number, customer first and last name, middle initial: string data type
- Opening balance: decimal data type
- Transaction date: date data type

### Tangibles

Follow Prof. Scanlan's instructions for the first assignment when submitting this assignment. An evaluation form will be made available at a later date.

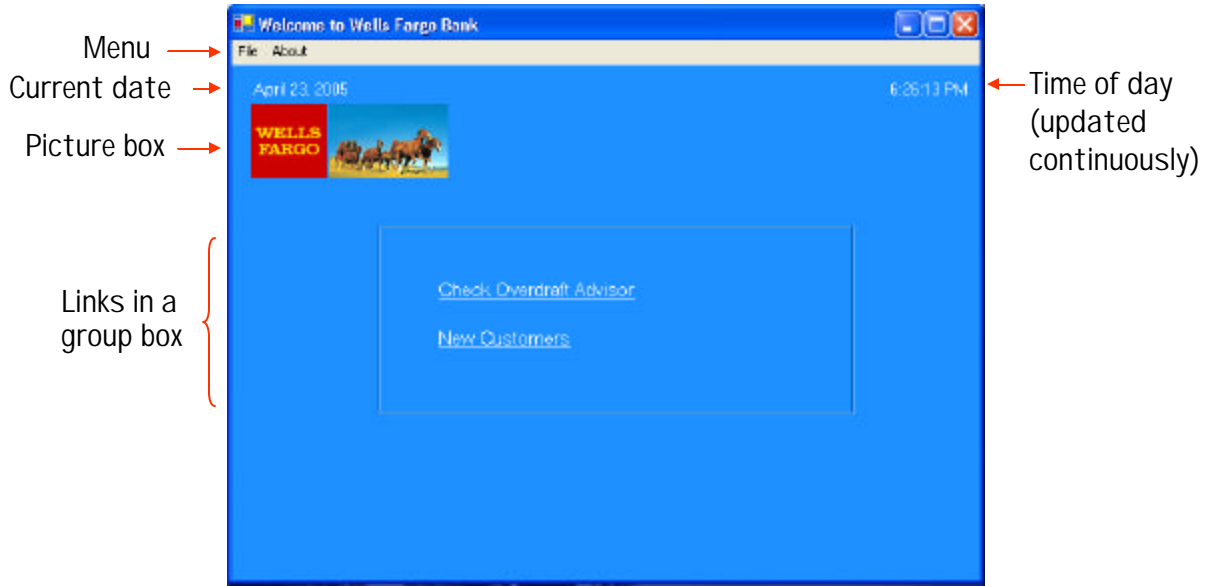


Figure 1. Main menu form

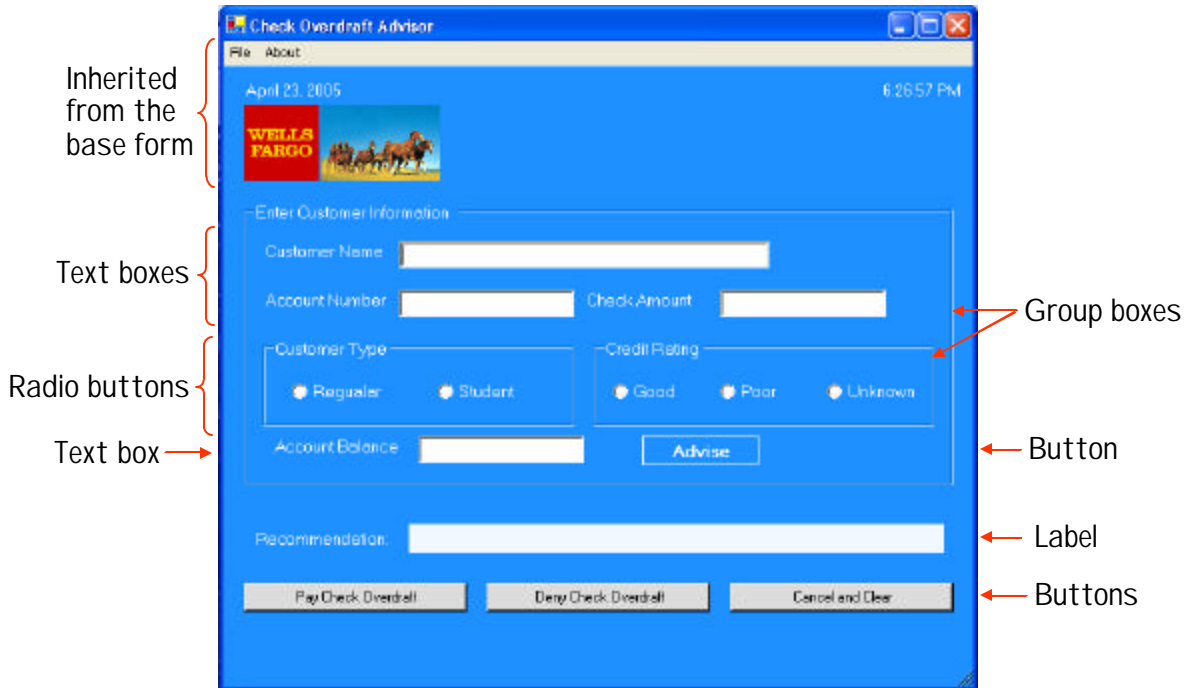


Figure 2. Check overdraft form

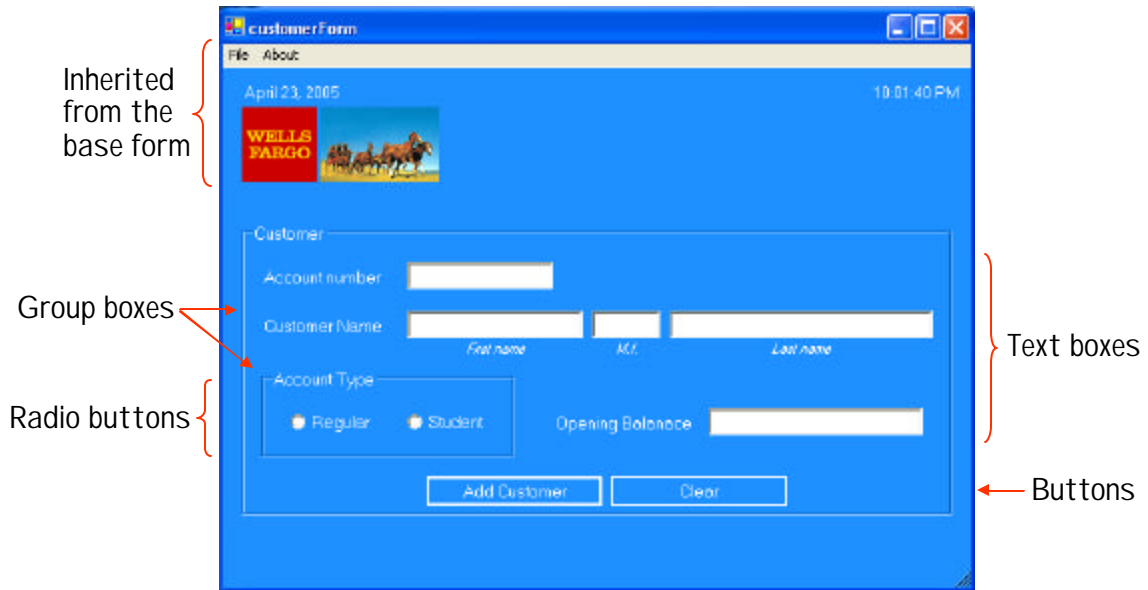


Figure 3. New customer form