Background

For this assignment you will be writing a Visual Basic 6.0 computer program which manages a DVD club called **Watchflix Club**. This DVD club is similar to the popular Columbia House music club, but much simpler.

Your program will accept DVD order information from new customers, one customer at a time. It provides a sales receipt for each new customer's purchases, and also has the ability to show summary information for all customers who have placed orders.

When the program begins, the user starts by entering the customer's name, and how many DVDs of each type this customer would like to order. There is information below about the costs of the two different types of DVDs and how many free DVDs of each type the customer is entitled to. The customer indicates how much s/he pays towards the order, and then clicks to place the order. At this time a receipt is printed on a label, indicating if the customer owes any more money or has a credit.

The program repeatedly accepts order information from customers and prints a receipt for each order. At any time, a summary can be requested by the user that displays various totals for all customers.

Your assignment

You will create a Visual Basic 6.0 project that contains a single form that has text boxes, command buttons, and labels. Please keep "ease of use" and "good design" in mind as you design your form. The behaviour of each command button on your form is described below. For each command button, you will write the Visual Basic code required to carry out the indicated task. Keep "good programming style" in mind as you write your code.

Form

The form includes the 7 command buttons described below. Your program will use the following 4 text boxes for data input: the customer's name, the number of silver level DVDs ordered, the number of platinum level DVDs ordered, and the amount paid towards the order.

The program uses a single label to display either the receipt that is printed when the user clicks on the **Place Order** button or the summary information that is printed when the user clicks on the **Summary** button. You are welcome to use other labels to give your program a nice appearance and to make it easier to use.

All error messages generated by your program will be displayed using message boxes. Your program only needs to detect the input errors that are outlined in this assignment description (only the **Place Order** button has input error checking).

Place Order Button

After the user enters the customer's name, the number of silver level DVDs ordered, the number of platinum level DVDs ordered, and the amount paid towards this order, the user must click on the **Place Order** button. When the **Place Order** button is pressed, the program uses a label to display the customer's receipt. The program also registers information into the various totals that are stored to be used for the summary (this uses global variables).

When the **Place Order** button is pressed, the program must use a message box to display an error message if the user has made a data entry (input) error. Implementing input error checking is worth 1 mark out of 10.

- The program will display an error message if no customer name has been entered.
  After the error message is displayed, the focus should be set back to the customer's name text box.

- The program will display an error message if the number of silver level DVDs, the number of platinum level DVDs, or the amount paid is not a number. (Note that the user must enter zero, not blank, if no silver or platinum DVDs are purchased.)
  After the error message is displayed, it clears the silver, platinum, and amount paid input text boxes and sets the focus back to the input text box for the number of silver level DVDs.
- An error message is displayed if the value entered for the number of silver level DVDs, the number of platinum level DVDs, or the amount paid does not make sense (but for the number of silver and platinum level DVDs, you can assume that the user will never enter a number with a decimal point, such as 12.5, and therefore you don't need to check for this).
  After the error message is displayed, it clears the silver, platinum, and amount paid input text boxes and sets the focus back to the input text box for the number of silver level DVDs.

In the case that there is more than one data entry error entered by the user, the program only reports the first data entry error it finds (hint: use an If...ElseIf statement -- this will be discussed at the end of your W6 lecture). When there is a data entry error, the receipt is not displayed and the information for this order will not be included in the summary (it is as if the transaction was cancelled and did not take place).

If there are no input errors, the **Place Order** command button should display a purchase receipt with the following information: customer's name, number of silver level DVDs, number of platinum level DVDs, amount paid, total cost of the order, and the amount the customer owes or has as a credit. Yes, it is possible for a customer to overpay an order -- in this case the customer would have a credit, instead of owing money.

Here is how to determine the total cost of the customer's order. A customer can receive up to 5 free DVDs, at most 2 of which can be platinum level DVDs. After the free DVDs, the customer is charged $19.99 for each additional silver DVD ordered, and $24.99 for each additional platinum DVD ordered. There is a $5 shipping charge for every order, and also 6% GST and 8% PST (the tax is applied to the DVD cost and to the shipping cost). Note that the shipping charge and taxes apply even if the total cost of all DVDs ordered is $0.

- For example, if a customer has ordered 7 silver DVDs and 3 platinum DVDs, the customer would get a total of 5 DVDs free (3 silver DVDs and 2 platinum DVDs). The customer would be charged $19.99 each for the 4 additional silver DVDs and $24.99 each for the 1 additional platinum DVD, plus $5.00 for shipping, and 14% taxes, which would be a total of $125.34.
- Here is another example. Suppose a customer orders 6 silver DVDs and no platinum DVDs. Again, the customer will get a total of 5 DVDs free, but in this case all 5 will be silver DVDs.
- What would happen if a customer orders 2 silver DVDs and 1 platinum DVD? The customer in this case would not need to pay for any DVDs, but would be charged sales tax on the shipping cost.
- What if the customer orders 2 silver DVDs and 5 platinum DVDs? Since the customer can receive <u>at most</u> 2 free platinum level DVDs, the customer would only end up getting 4 free DVDs (2 silver and 2 platinum). The customer would have to pay for the 3 additional platinum DVDs.

Hint: In addition to creating variables to store the number of platinum DVDs ordered and the number of silver DVDs ordered, you may find it helpful to keep track of the maximum number of free DVDs of each kind that the user will claim. Keep in mind that you will need to determine the number of free platinum DVDs first in order to be able to determine the number of free silver DVDs. It is HIGHLY recommended that you work through your solution on paper first!

After you have calculated the cost of the DVDs, you must determine if the customer owes money or has a credit. Use the amount paid entered by the user and your calculated total cost of the order to determine this.

When you print the information for the current customer on the label, you will either write the amount owing or the credit amount (but not both the words 'owe' and 'credit'). The actual amount will always be shown as a positive value, so the word 'owe' or 'credit' must be included to indicate if the cost of the order is positive or negative. For

example, your label could display either "Credit of $15.27" or "Owes $64.22" (notice that neither number is displayed with a negative sign).

After displaying the receipt for this customer, the program will need to update its global variables. Think carefully about which values will be needed in the summary to determine which information needs to be stored.

Clear Order Button

The **Clear Order** command button clears the information in all text boxes as well as the label that displays the receipt/summary information. Pressing this button also resets the text showing on the **Encode/Decode** button so that it says "Encode" (this can easily be done by assigning a value to the Caption property of this command button).

The **Clear Order** button is often pressed after one customer has made a purchase, and you are about to enter the order information for the next customer. In fact, after pressing the **Place Order** button, the focus is set to the **Clear Order** button. After pressing the **Clear Order** button, the focus is set to the customer's name text box.

Summary Button

The **Summary** command button prints summary information on a label. The summary is for all orders that have been placed since the program was started. The summary information must be displayed on four separate lines within the label. The summary information to be displayed must include:

- Total number of customers entered so far.
- Total cost of all DVDs ordered (not including shipping or tax)
- Total amount owing or credit amount for all customers (either show amount owing or credit amount as a positive number).
- Average order cost (includes cost of DVDs, shipping, and tax) per customer, rounded to two digits to the right of the decimal point.

Hint: use the values that are stored in your global variables to obtain the information that you need to display for the summary. What needs to be displayed in the summary will help you to determine which global variables your program will need.

Encode/Decode Button

Implementation of the Encode/Decode button is worth 1 mark out of 10 on this assignment. Please note that this is only one button that has either the word "Encode" or "Decode" displayed on it at any given time.

You do not need to do any error checking at all for this button. You can write your code assuming that whenever the **Encode/Decode** button is pressed, the customer name text box will always contain a name that has a first name, a space, and a last name. For simplicity, assume that the customer name text box will never contain more than one space.

The **Encode/Decode** command button doesn't have any useful purpose other than to give you practice with using string functions and using a loop (since for full credit you MUST at least use one loop for this button -- it will likely be a loop that goes through each character in the customer's first name, one character at a time).

Initially this button will have the word **Encode** displayed on it. When the user presses the **Encode** button, it encodes the name that is currently in the customer name text box by reversing the characters in the customer's first name. For example, if the name "Sarah Sivarajah" was in the customer name text box and the user clicked on **Encode**, the name in this text box would change to "haraS Sivarajah" and the button would change to **Decode**.

Once the **Encode** command button is clicked, the name that was in the customer name text box is replaced by this new encoded name, and the text showing on the command button is changed so that the command button now displays the word **Decode**. By pressing on the **Decode** button, the first name is again reversed so the name in the text box changes back to its original form, and the word showing on this command button changes back to **Encode**. You can continually click on the **Encode** and **Decode** buttons to change the name back and forth between encoded and decoded form. The focus stays on this button when pressed.

Hint: The InStr function will be helpful for determining the position where the first name ends and the last name begins.

Here is another example. If "Arnold Swartzenegger" was entered in the customer's name text box, and the user clicked on the "Encode" button, the name in the text box would change to "dlonrA Swartzenegger" and the text displaying on this button changes from "Encode" to "Decode". When the user clicks on the "Decode" button, the first name is again reversed so that the name in the text box would change back to "Arnold Swartzenegger".

About, Exit, Print Buttons

When the **About** command button is clicked, use a message box to display your name as programmer and the approximate date you completed the program. The **Exit** command button causes the program to exit and the **Print** command button prints the form to the default printer.

Smart Advice

You must insert the following line of code in the General/Declarations section of your program. This forces all variable names to be explicitly declared, reducing the possibility of inadvertently introducing a bug into your program. Option Explicit

Spend a lot of time thinking about your program before jumping into writing code. You may find it helpful to write pseudocode for each subprogram. It will be particularly important to write pseudocode for the more complex subprograms (such as the **Place Order** command button).

It may be helpful to write the code for the **Place Order** subprogram first. Initially write this code so that it does not handle the special cases that generate input error messages. Once the program is working correctly for input without errors, go back and add the code that will display error message boxes when invalid input is entered.

Think carefully about which global variables you will need to store in this program. Where are these variables incremented and where are they used? Only shared variables should be in the General/Declarations section.

You may want to leave the implementation of the code for the **Encode/Decode** command button until after you have everything else working. This part of the program will not affect the execution of the rest of your program, so it does not have to be coded at the same time. You will find the String functions that were discussed in lecture and tutorial useful for this.

Submission

The following is a list of what you need to submit electronically for this assignment.

- Project file (use a2.vbp for the filename)
- Form file (use a2.frm for the filename)
- Executable file (use a2.exe for the filename)
- StarOffice Writer file (use snapshots.odt for the filename) containing snapshots that show the contents of your form at various points in the execution of your program. How and when to get the snapshots is described below. Please put them into the right sequence.

These four files will be submitted using Blackboard. Log into Blackboard (http://portal.utoronto.ca) by entering your UTORid and password. Follow the link on that web page for setting up a UTORid, if you don't already have one. Once you are logged in,

- click on the link for the **CSCA02** course
- click on **View/Complete Assignment: Assignment 2**
- if you haven't completed all of this assignment or if your assignment is not working correctly, please leave a note for your TA in the **Comments** box, explaining the limitations of your program.
- next to **Attach local file**, click on **Browse...** to select your first file and then click the **Add Another File** button to add this file. For each additional file, repeat this process (i.e., click on **Browse…** and then **Add Another File**). The name of each file to be submitted will be shown in a list.
- make sure all of the files that you want to submit are listed before clicking **Submit**, as you **CANNOT RESUBMIT AN ASSIGNMENT** or individual files for an assignment! If you are not ready to submit your files, just click on Save and visit this web site again later to submit.

Always be sure to make a backup of your submitted files, just in case something goes wrong.

Instructions for preparing snapshot file
Start the program once, and then continue running the program after each snapshot you create. Create your snapshot file by clicking the ALT-PrintScreen buttons and pasting each snapshot into a single StarOffice Writer document (your TA will explain this further).

- Start your program. Enter *Aunt Jemima* for the customer's name, *7* as the number of silver level DVDs ordered, *3* as the number of platinum level DVDs ordered, and *45.00* (notice that the dollar sign is not typed when we enter a money amount) as the amount paid by this customer. Click on the **Place Order** button. At this point, capture snapshot #1, which will display the receipt for this customer.
- Click on the **Clear Order** button and then enter *Pippi Longstocking* for the customer's name, *6* as the number of silver level DVDs ordered, *0* as the number of platinum level DVDs ordered, and *65.00* as the amount paid. Click on the **Place Order** button. At this point, capture snapshot #2, which will display the receipt for this customer.
- Click on the **Clear Order** button and then enter *Huckleberry Finn* for the customer's name, *2* as the number of silver level DVDs ordered, *5* as the number of platinum level DVDs ordered, and *85.27* as the amount paid. Click on the **Place Order** button. Then click the **Encode** button. At this point, capture snapshot #3, which will display the encoded name and the receipt for this customer.
- Click on the **Clear Order** button and then press the **Summary** button. At this point, capture snapshot #4, which will display the summary for all customers.