For this assignment name the project/solution A5-FName and give the form/file an appropriate name. When you have completed the project, zip the top-level folder (with all its contents) and attach the resulting file to an email sent to cs405@cs.unh.edu (put 405-A5 in the Subject: line, and change the extension to .zip if necessary). Although it’s a good idea to create a test plan for any application you create, you are not required to submit a test plan for this or any subsequent assignments. The assignment is officially due by midnight on the date indicated above right.

The project is intended to allow the user/clerk to enter data or make selections to calculate an employee’s pay. Your program should present a user interface similar to the one below, with startup values identical to the first sample form displayed below. Your code as always must follow the details of the Coding Specifications handout.

The basic idea is for the clerk to type in the name of the employee, set the two scroll bars to the employee’s current pay rate and the number of hours worked that week, and set Dependents, Status, State Tax and Retirement appropriately for that employee. Whenever any of these settings are changed, the Gross, Deduction, and Net values at the top should change accordingly.

The Hours scrollbar should range from 0 to 50 hours, in small increments of .25 hours and large increments of 1 hour. The PayRate scrollbar should range from 8 to 30 dollars per hour, but in small increments of $.10 and large increments of $1. Please note that VB only allows integer values in its scrollbar properties, so you will have to do the decimal values in code. To get the maximum value of the scrollbars to work properly, you will also have to set the Maximum property to the value required (N) + LargeChange – SmallChange.

If the worker is a Consultant (rather than a regular Employee), then no Deduction should be taken (and all the other radio buttons and check boxes should be turned off). For an Employee, no Dependents (None) means a 15% deduction, One means a 12% deduction, and Two means a 9% deduction. In addition, if the Employee is liable for State Tax that means a further 4% deduction, and if in the Retirement plan another 10% is deducted. (Note: all deductions are taken from the Gross, not from the calculated Net)

The Display button should cause the appropriate payroll information to appear in the Message Box. This data should be neatly formatted, using multiple lines and tabs (the ControlChars constants will be useful here). Also, take some appropriate action if the user tries to Display the data without having entered an employee name.

The Reset button should cause the form/window to return to its original settings (i.e., appear exactly as it does when your application starts up - see first sample form below). Make sure the tab order is reasonable, that the Display button functions as the Default/Accept button, that the Reset button functions as the Cancel button, and use Access Keys where appropriate.

Programming note: You should use at least one general procedure somewhere in your code. The general procedure(s) does(do) not need to have any arguments. You should also use named constants wherever appropriate.
Startup/Reset Setting:

Sample settings/output:

MessageBox when Display button clicked for above: