**Your Name:** Jeremy Stroud, Kameron Wagner, Gary Sweet, Peter Moore, Lucas Henn, Victor Short

**Pseudocode Plan for** Group Project

|  |  |  |
| --- | --- | --- |
| **Object (call)** | **Method (event)** | **Action - Steps** |
| Windows button | btnExit (Click) | Close the form |
| Windows button | btnClear (Click) | Enable grpCustomer  Clear Customer fields (txtName, txtAddress, txtCity, txtState, mskZip)  Reset chkNewCustomer to default  Clear Item fields using ResetItemFields()  Reset Radio buttons to default (Shipping and Payment Type)  Clear display fields (txtAmountDue, lblDiscount, txtSalesTax, lblTaxRate, txtShippingAndHandling, txtTotalAmountDue)  Set lblDiscount and lblTaxRate to Visible = false  Set accumulators (decRunningTotal, intRunningQuantity) to zero  Disable btnSummary  Set focus to txtName |
| Windows button | btnAddOrder (Click) | Declare local variables (decimal decAmountDue, decPrice; int intQuantity)  Validate user input fields using ValidateFields()  If ValidateFields() is true:  Disable grpCustomer  Convert input txtQuantity to Integer, assign to intQuantity  Convert input Price to decimal, assign to decPrice  Calculate Amount Due (intQuantiy \* decPrice), assign to decAmountDue  Add decAmountDue to global decRunningTotal  Add intQuantity to global intRunningQuantity  Reset Item fields using ResetItemFields()  Set focus to txtDescription  Enable btnSummary |
| Windows button | btnSummary (Click) | Declare local variables (decimal decSalesTax, decTaxRate, decShippingHandling, decTotalDue)  Get Discount using GetDiscount() and subtract decDiscount from decRunningTotal  Assign Tax Rate to decTaxRate using GetTaxRate()  Calculate Sales Tax due (decRunningTotal \* decTaxRate) and assign to decSalesTax  Get Shipping and Handling fees using GetShipping () + GetHandling() and assign to decShippingHandling  Calculate decTotalAmountDue (decRunningTotal + decSalesTax + decShippingHandling)  Display decAmountDue in txtAmountDue formatted as Currency  Display decSalesTax in txtSalesTax formatted as Currency  Display decShippingHandling in txtShippingHandling formatted as Currency  Display decTotalAmountDue in txtTotalAmountDue formatted as Currency  Display message box based on shipping selection using CalculateDeliveryDate() |
| RichTextBox | rtbShipperWebsite (Click) | Open Internet Explorer to shipper website |
| Form | frmMailOrder (Load) | Set txtDate to current date, display using month/day/year |
|  | ResetItemFields() | Clear Item fields (txtDescription, txtQuantity, txtPrice) |
|  | GetDiscount() | Declare constant decimal decDISCOUNT\_RATE and assign 10%  If chkNewCustomer is checked, set lblDiscount to discount rate and visible then return discount amount (decDISCOUNT\_RATE \* decRunningTotal)  If chkNewCustomer is not checked, clear lblDiscount and hide, return 0 as discount amount |
|  | GetTaxRate() | Declare constant decimal decTAX\_RATE and assign 8.7%  If txtState is “WA”, set lblTaxRate to decTAX\_RATE and display, return decTAX\_RATE  If txtState is not “WA”, clear lblTaxRate and hide, return 0 as tax rate |
|  | GetShipping() | Declare and initialize constant decimal values ( decEXPRESS\_SHIPPING = 13.25m,  decGROUND\_SHIPPING = 5.75m)  Declare and initialize decimal values (decShipping) to 0  If radShippingExpress is checked, set decShipping to decEXPRESS\_SHIPPING  Else if radShippingGround is checked, set decShipping to decGROUND\_SHIPPING  Return decShipping |
|  | IsValidState() | Checks string against list of state abbreviations, returns true if match is found |
|  | IsPresent() | Returns true if field contains data |
|  | IsDecimal () | Returns true if field contains a valid decimal value |
|  | GetHandling() | Declare and initialize constant decimal values (decHANDLING\_FEE01 = 1.50m,  decHANDLING\_FEE02 = 2.75m,  decHANDLING\_FEE03 = 5.50m)  Declare and initialize decimal values (decHandling) to 0  If intRunningQuantity is less than or equal to 2, set decHandling to decHANDLING\_FEE01  Else if intRunningQuantity is less than or equal to 4, set decHandling to decHANDLING\_FEE02  Else if intRunningQuantity is greater than 5, set decHandling to decHANDLING\_FEE03  Return decHandling |
|  | ValidateFields() | Clear Error Provider  Set dictionary of form controls and friendly names  For each control in dictionary, check if value is valid using IsPresent(), IsValidState(), and IsDecimal() as appropriate  Return true if all fields validate successfully  If any field is not valid  Set Error Provider message using friendly name and defined string, set focus to invalid control, if control is a textbox, select all text currently in textbox, and return false |
|  | CalculateDeliveryDate() | Set DateTime to today’s date and time  Add Int parameter to DateTime and return result |