

# Applications Programming

Programming for Userform Events

# Setting the Tab Order

- The tab order is the order in which the focus moves from one essential control in a dialog box to the next essential control as you press the Tab key. An essential control is one that can receive input from the user.
- The first essential control in the tab order typically is located in the upper-left area of the dialog box.
- The tab order is determined by the TabIndex property of the controls included in the dialog box. When you add a control to a form, the control's TabIndex property is set to a number that represents the order in which the control was added to the form. The control whose TabIndex value is 0 will receive the focus first, because it is the first control in the tab order.

# Keyboard Access

- Providing keyboard access to the controls in a dialog box allows the user to work with the dialog box using the keyboard rather than the mouse. There are times when the user may have to or prefer to use the keyboard.
- You can provide keyboard access to a control by assigning an accelerator key to the control. An accelerator key is usually shown as an underlined letter and is used in combination with the Alt key as a shortcut for selecting a control.
- In all Microsoft Office applications but Access, you use a control's Accelerator property to assign an accelerator key to the control. In Access, you place an ampersand to the left of the appropriate letter in the control's Caption property.

# Default/Cancel Properties

- The *Default Button* is the one that is selected automatically when the user presses the Enter key, even when the button does not have the focus. A command button is made the default button if its Default property is set to the Boolean value True.
- The *Cancel Button* is the one that is selected automatically when the user presses the Esc Key. A command button is made the cancel button if its Cancel property is set to the Boolean value True.

# Event Driven Programming

- What is an event?
  - Actions performed by the user, such as clicking, double-clicking, and scrolling, are called events.
- What is event driven programming?
  - We develop event procedures to tell the objects how to respond to an event.
- What is an event procedure?
  - Event procedures are subroutines that run in response to an event rather than in response to running a macro.

# Examples

- Clicking a button generates a `CommandButton_Click` event for which you may write code to execute when this event occurs (i.e., when the button is clicked).
- Some actions generate multiple events (i.e., `SpinButton`'s `Spinup` and `Change`) and you may write code for any or all of them.
- To find out what events are associated with controls you have on the form,
  - Double click the control to activate code module;
  - Click drop-down list in upper right.

# Design Checklist

- Are similar controls the same size?
- Are the controls evenly spaced?
- Can every control be accessed with a hot key?
- Is the tab order set correctly?
- Will your VBA code take appropriate action if the dialog box is cancelled or the user presses Esc?
- Are there any misspellings in the text?
- Does the dialog box have an appropriate caption?
- Are the controls grouped logically (by function)?

# Example

- Ask the user to enter the following information (using InputBox)
  - the number of students that are 12 years or older;
  - the number of students that are 5 to 11 years old;
  - the number of accompanying adults (including teachers, and bus drivers if there are any) travelling with the students;
  - whether a bus is used, and if a bus is used, the length (in feet) of the bus used by this group.
- Calculate the total charge the school needs to pay to transport this group from Departure Bay to Horseshoe Bay via BC Ferries according to the information collected from the user and the fare information stored in the worksheet.