# CE 05: Custom Event Arguments

# Setup:

You will be creating a new Visual Studio project named "FirstnameLastname\_CE05" using the C# Windows Forms Application template.

Make sure you include your name, class and term, and code exercise name and number as comments to the top of each code file of your project. You will lose 5 points automatically if these comments are missing.

# Grading:

Refer to the CE: Custom Event Arguments tab of [this](https://docs.google.com/spreadsheets/d/1X3ao-O9B5rJMavuXt8w5SydRPoxa1mIwOJ8opVNL1ag/edit?usp=sharing) rubric to understand the grading for this assignment.

# Deliverables:

You will compress and upload a file named "FirstnameLastname\_CE05.zip" with the following contents with the following names:

* Project - Folder containing your entire project and all files necessary to build the project(csproj, .cs files, App.config, and Properties folder).

Be sure to upload the correct project and all required files the first time as only one submission will be allowed. No extra time or consideration will be given if the wrong files are uploaded.

# Instructions:

#### **Topics:** hiding/showing controls, StatusStrip, custom event arguments, radio buttons

**Images Link**: Click [here](https://drive.google.com/open?id=0BykzKaX7lsyMbTJydk8zUXdoWW8) to get the images you’ll be using for this code exercise.

Students will create an application that uses a dialog to create a spaceship object to populate a ListView. This ListView will be able to display in either large or small icon mode using a menu option to switch between them. Selecting a spaceship in the ListView will give the user the ability to modify that object’s values and apply those changes.

You will be creating two forms, the main window with a ListView, MenuStrip, and StatusStrip, and a dialog (i.e. modal) window with some user input fields for making a spaceship object. The inputs needed for the spaceship are a TextBox for name, a numericUpDown for crew size, a checkbox for indicating active duty, and a set of 3 radio buttons for selecting the type of ship (Cruiser, Destroyer, or Freighter). There should be three buttons on this dialog: OK, Cancel, and Apply. OK and Cancel will always be visible, but the Apply button will only appear when attempting to modify an item from the ListView. The OK button will add a ListViewItem with the current input to the main window’s ListView, and the Cancel button will close the window without changing anything in the main window. The Apply button should use a custom EventHandler and custom event arguments to modify the selected item in the main window’s ListView to have the current input values in the dialog. The main window’s MenuStrip should have three separate menus: File, Ship, and View. Each of these menus will have the following options: File->Clear to clear the ListView, File->Exit to exit the application, Ship->New to open a dialog to make a new spaceship without the Apply button being visible, View->Large to make the ListView use LargeIcon mode, and View->Small to make the ListView use SmallIcon mode. The currently selected icon mode should have a checkmark displayed next to it. The StatusStrip should appear at the bottom of the window and have a single ToolStripStatusLabel showing the current number of items in the ListView. There should be unique images for each of the 3 ships with a large 32x32 and a small 16x16. Double-clicking an item in the ListView should bring up the spaceship dialog with the Apply button visible and the fields pre-populated with the selected item’s current values.

Follow these guidelines for this application:

**User Input Dialog**

* Radio buttons are used for ship selection that function as radio buttons should (only one option selectable at a time).
* Apply button should only be visible when opened by double clicking a ListViewItem.
* A TextBox for name, NumericUpDown for crew size, and CheckBox for active duty are present for additional user input about the spaceship.
* OK and Cancel buttons should always be present on this dialog.

**ListView/Main Window**

* ListView exists, set to dock/fill the parent container, and can be toggled between LargeIcon and SmallIcon mode.
* A StatusStrip is present at the bottom of the window that always displays the current Item count of the ListView.
* Double clicking a ListViewItem brings up the spaceship dialog as does selecting the Ship->New MenuStrip option.
* 3 Unique images are used in the ListView that correspond to the three ship types. Large(32x32) and Small(16x16) image lists are used for this purpose.

**Events**

* When the Apply button is pressed, a custom EventArgs class is used with an EventHandler to pass the updated spaceship information to the main window. The updated spaceship fields are used to update the currently selected spaceship.
* When the OK button is pressed, a new spaceship with the current input values is added to the ListView and the spaceship dialog is closed.
* When the Cancel button is pressed, the spaceship dialog closes without affecting the main window’s controls.

**Extra Information**

Go back through your code and check for the following:

* Your application must compile and must not crash when starting up.
* All variables and methods are named appropriately.
* Input controls are appropriately labeled.
* Any information being output to the user should be clear and concise.
* Make sure nothing accesses an object that doesn’t exist.
* Add comments to your code to indicate that you know what each line does.