The Visual Basic Editor is used to manage projects and to view the code generated by recording or writing macros. Procedures are created by typing VB code directly into the Code Window of the Visual Basic Editor which is shown in the figure below. The three ways to access the editor are: pick the Visual Basic Editor button from the Visual Basic toolbar, select the menu sequence Tools – Macro - Visual Basic Editor or press Alt-F11.

All of the windows within the VB Editor can be moved around in the editor window. They can all be docked in different locations, with the exception of the code window which is undockable. The windows can be turned on or off using the View menu. The common parts of the window are:

- **Project Explorer**: an organized view of the components of a project
- **Properties Window**: the place to change and set the attributes of objects
- **Code Window**: displays VB code, which is written and edited here
- **Immediate Window**: allows for the execution of individual lines of code and obtaining the value of variables
- **Locals Window**: displays the value of variables and properties during code execution
- **Object Browser**: lists all available objects, including their properties, methods and events
VB Editor Menu Bar:
The menus found on the menu bar can be used to perform most of the development tasks in VBA. A brief explanation of the menus follows:

<table>
<thead>
<tr>
<th>Menu</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>saving and printing functions for code and forms</td>
</tr>
<tr>
<td>Edit</td>
<td>undo, cut, copy, paste, indent code, clear, select and find</td>
</tr>
<tr>
<td>View</td>
<td>view code, windows and Object Browser</td>
</tr>
<tr>
<td>Insert</td>
<td>insert procedures, forms and new modules</td>
</tr>
<tr>
<td>Format</td>
<td>format form objects</td>
</tr>
<tr>
<td>Debug</td>
<td>tools for debugging or finding errors in a procedure</td>
</tr>
<tr>
<td>Run</td>
<td>run, break or stop the execution of a procedure for testing</td>
</tr>
<tr>
<td>Tools</td>
<td>access project properties and other functionality</td>
</tr>
<tr>
<td>Window</td>
<td>arrange windows within the VB Editor</td>
</tr>
<tr>
<td>Help</td>
<td>access VB help files</td>
</tr>
</tbody>
</table>

Code Window:
The code for procedures (and recorded macros) is viewed and can be edited in the code window. Individual procedures are separated in the code window by solid gray lines. Every macro begins with the line `Sub MacroName` and ends with the line `End Sub`. The code is executed one line at a time from top to bottom. Execution ends when the End Sub line is reached.

If the code window is not displayed in the VB Editor, right-click on the module in the Project Explorer and select Show Code.
Procedures can be executed by placing the cursor within its procedure separators and pressing the F5 key. Pressing the F8 key when the cursor is in the code window activates the *Step Into* mode of program execution. This method executes one line of code each time the F8 key is pressed. This is a good method to use when debugging procedure code.

When you move the focus off of a line of code, the code formatter automatically does the following:
- places keywords in proper case
- adjusts spacing
- adds punctuation
- standardizes variable capitalization

Text in the code window is automatically formatted by VBA as follows:

Green = Programmer’s comments used to document what code in the procedure does. Comments are not executed by the computer when the procedure is run. Any line starting with an apostrophe (‘) is treated as a comment.

Blue = VBA keywords, such as Sub, End Sub, If, End If, With, and End With.

Black = All other text.

**Editing Guidelines:**
It is considered good practice to follow the following guidelines when writing code:
- Indent text within control structures. To do this, select one or more lines of code and press the Tab key. To remove an indent, press Shift-Tab.
- Complete statements by pressing the Enter key or by clicking somewhere else in the window.
- Include comments about the code by starting the line with an apostrophe (‘). Comments are ignored by the computer during the execution of code.
- Statements can be carried over to the next line by typing a space and underscore character at the end of the line. This also works for comments. Strings that are carried over to the next line require a closing quote, and ampersand (&), and a space before the underscore.
- Multiple commands can be placed on a single line if separated by a colon (:). This method is usually harder to follow when reading code.

**HELP!**
Within the VB Editor, you can click on words and objects and press the F1 key to access context sensitive help files. Help is also available from the Help menu.