Grading Rubric

The things I look for when grading are very simple. The presence and the quality of planning documents, the naming of files, the layout of the forms, the arrangement of the code, the inclusion of all the pieces. These are the basics.

The planning documents are more important from Chapter 5 to the end of the book . You have to know what the end product is going to do and look like early on. Design is the second step in the process and that's not just layout design that includes the design of the logic and the code that comprise the solution.

ADDIE

Analyze: Break the problem down into its most discrete components and solve each one.

Design: Design the look of the forms and the logic for the solution.

Develop: Develop the code.

Implement: Run your program. Test the results.

Evaluate: Determine what worked and analyze the solution for what didn't. Repeat until done.

Naming Conventions

Project Names

Naming the projects as described in the Syllabus using the format:
LastName+ProjectType+ChapterNumber.

	Correct	Incorrect
Examples:	MartinCPA9	Martin_CPA_9
	ThompsonGPD7	ThompsonAssignmentGPD7

There are reasons for the naming convention and it's part of the requirement.

Form Files

Form files such as Form1.vb should always be renamed with an appropriate description. For example Form1.vb of a calculator program should be renamed Calculator.vb. Form file names should always include ".vb" or they will not function correctly. Form files should not include a three letter prefix.

Form Object

Form objects are different. Form objects are where you place controls such as buttons, labels, or text boxes. Form objects should be renamed with the prefix "frm". In the previous example the form object might be renamed to frmCalculator.

Controls

All controls such as buttons, labels, text boxes, etc., should be renamed with a descriptive name including the three character prefix as described in Appendix D, page APP 24, Figure D-3.

Renaming Files

Project files may be renamed only while inside Visual Studio. Visual Studio will manage the referenced relationships of the project's files. Files renamed outside of Visual Studio using Windows Explorer will break the references and prevent the program from running.

Form Design

The Look

Make your forms look nice without being gaudy. A nice graphic related to the topic can add clarity to your purpose. Graphics over a background image could get confusing and start to look like a circus.

Use color to enhance the appearance of your forms. Color should be used to draw the user's attention to objects that require their interaction.

Text Property

Set the Text Property of your form. Form1 in the title bar is not professional.

Icon

Use an appropriate icon on your forms; set the icon property of the form. There's an icon library on our website: http://www.cop1000.com/Links.aspx. It's the indecipherable picture that's supposed to be books in a library. Check the tooltip.

Code

Comment Header

There should be a comment section at the top of the code page that lists your project name, your name, the date you started the project, and the program's purpose.

Options

Option Strict On and Option Explicit On should be at the top of every code page.

Empty

Do not leave empty event handlers in your code.

Arranging Your Code

Put the frmYourForm_load sub, if you have one, at the top of the code. Put the Exit sub at the bottom of the code. All other subs and functions should be in alphabetical order between the load and exit subs.

Whitespace

Use no more than one blank line to separate code sections. Be consistent in formatting your code. If you use a blank line after the signature line of a Sub, use one blank line just before the End Sub statement.

Functional Comments

Use comments to explain how and why you write segments of code in a particular way. Not a book, just straight to the point.

Planning Documents

General

Planning documents are for chapter projects and not Guided Program Development projects. There are three required planning documents; Requirements Document, Use Case Definition, and the Event Planning Document. All are required after Chapter 3. The quality is expected to be good and the content functional. The three planning documents are 20% of the project grade.

Requirements Document

This is your notice of intent to associate programmers or the test and maintenance teams. It tells them the title, the purpose, and the method of operation for your program. It lists and describes all calculations, restrictions, the intended environment, and configuration required for your program to run. Any additional notes to that will aid in the operation of the program should be included.

Use Case Definition

The Use Case Definition tells others how to run your program. It should contain list of the sequence of events the user must take to run the program and it may contain screenshots from the program itself.

To make screenshots:

1. Pick the window to be shown.
2. Hold down the Alt key and press the PrtScn key (Alt + PrtScn).
3. Put your cursor in the document where you want the image to reside.
4. Hold down the Ctrl key and press the V key (Ctrl + v).

Event Planning Document

A table listing the events to be programmed to complete the functionality of the solution. For each event there should be a purpose or task which the event will satisfy, a description of the object used to trigger the event, a name of the event used, and a list of the processes that should occur when the event fires. To communicate with developers and teams it helps to list the properties and their settings. For management and administration purposes, it helps to list the estimated and actual time required for adding each item in the list.

Building Blocks

Accept or Cancel

The use of the form's AcceptButton and CancelButton properties. We've seen how they're used so it is my expectation that you should use them as often as it is appropriate in your programs.

Tab Order/Tab Index

Our author doesn't do this justice so I want you to learn to find things on your own. You can use the Help function, another book, or the Internet but when your user presses the Tab key the results should be logical, predictable, and controlled.

Splash/About

After we cover Splash and About screens, every project should include a SplashScreen and an About box.

The SplashScreen should announce the loading of your program and should run for 3-4 seconds.

The About box should be associated with a pull-down menu and contain a digital photograph of the developer and a two paragraph biography.