**General guidelines**

When receiving a new assignment, it is probably in your best interest to   
  
1. Place a printed copy of all the sample programs in front of you on a desk. Review these until they make sense. If you have not done so, create your own program that is similar to each sample provided.  It is crucial to gain experience and a solid foundation before starting a complex activity that demands accuracy such as computer programming. Starting a new task without preparation, only creates needless stress which is not healthy.

2. Read the assignment requirements, number the requirements and map each assignment requirement to the instructor-provided content that demonstrates the functionality.  
Also make a list of the required functionality - this becomes your plan of action.

3. With your 'mapping document' and plan of action you can then begin the assignment without the feeling that it is a 100% creative task. rather you are following best practices. After you complete and submit the assignment, you would be very smart to do the project again using similar functionality but your own scenario (perhaps building on a pet project, or working on a project that will turn into your final project for the class.

If you follow this structure, the difficulty (and time required) should plummet as you do not need to pull the code from memory in a trial and error approach (which is a rookie mistake). Rather the feeling is of doing a task that you are well-prepared for,

**Assignment overview**

You are asked to create the beginnings of a point of sale system for a bookstore in an .aspx webpage. The manager needs to know how much to charge for a transaction (the sales of a quantity of a book, magazine or toy), and also wants to know how much business is derived from each of these product categories; magazines, paperback books, and educational toys. The system is similar to those created in class, using one .aspx page. It makes sense to create a folder in your website named assignments, and place a new webpage into this folder called assignment1.aspx. This project gives you practice with input validation, using local variables, and using global variables.  
  
Note: You may undoubtedly want to make this program more advanced and more complex than is needed. Perhaps just keep it simple, since its the first assignment. After completion and submission, you are welcome to upgrade and complexify a new solution. There is no need for select case or looping methodologies for this assignment.

***How to Proceed***

a) First complete all programs shown or started in class. Then watch all videos related to the current content. If anything went wrong in class, troubleshoot, fix and understand the problems, talk to classmates and compare approaches. You do not want to be unsure of what to do. Starting an assignment without preparation is folly. "Failing to plan is planning to fail" (J. Wooden). Next read the program requirements (the required functionality)

b) write the input processing (calculations) and output steps in your native language on a piece of paper (plan your work)

c) sketch out the user interface (buttons, lists, textboxes, etc.) to make your plan

d) contact your homework buddy to make an appointment to get together and crunch out the project. Also start the project when it is assigned not the day before it is due.  
You can take the approach that one person types and the other helps problem solve (perhaps using a large screen TV), but be sure to switch roles. You will learn that it is no fun to code late

***Program functionality***

a) You are creating a point of sale program so that an employee at a bookstore can calculate how much to charge a patron for their order of one or more books, magazines or toys. This is a relatively simple introductory assignment so do not read too much into the instructions. The program should use a button click procedure to calculate and display the charge for the transaction including tax (use a 9.5% tax rate). Use textboxes to capture the price of the book and the quantity purchased. Use a radiobuttonlist (or dropdown list) with the three categories (books, toys, magazines) to indicate which category product is being purchased.

You are selling products that are in one of three categories (magazines, books, and toys). Calculate the price of the product times the quantity of the product sold, then add the tax.  Place a button on the form and double click it to get to the code page. you should see a button click procedure, your code goes in there. Use textboxes for data entry of quantity and price. Use a radiobuttonlist to select the category.  
  
b) Create three global decimal variables (Public shared) to keep a running total to count the # of units sold (depending on the product category - #magazines, #books, and #toys). So if a patron purchased some magazines you would increment the global variable for magazines (for example gdecMagazines += convert.todecimal(txtQuantity.text). Use a select case processing to control what variables are updated after each transaction  
  
c)  Create three more global decimal variables that get updated to keep a running total of the sum of the pre-tax revenue of transactions (depending on the product category - magazines, books, and toys).  
  
d) Use one textbox to show the price for the current transaction and a second textbox to show the running totals for each individual category (# transactions and total revenue). Use concatenation to build nice output.

e) include another button click procedure that clears the form as shown in the examples.

f) include input validation as shown in distributed examples to make sure data is entered into each required web control (for example stop the program if no category was selected or no quantity was specified.

***Program output***

c) Add a multi-line output textbox that shows the total to charge the customer (number of items purchased times sales price, then add in tax. Use concatenation.

d) add another output textbox that shows the manager the running totals for the day - number units sold and the before tax revenue for the day for each of the 3 categories - magazines, books, and toys.

e) format the webform nicely.

f) add an image control whose picture changes depending on what category was selected.

**Turn-In**

Note - please read the syllabus regarding grading of assignments.

Each assignment is worth 10 points total, however it is very rare for anyone to earn a full 10. Commonly grades are 8.5 - 9.25, with exceptional work earning higher points.   
  
1. For each assignment, we need 2 versions of your assignment   
a) upload a word-processing document with 1) a cropped screenshot of your code working.  2) your code  
The filename must have the format something like JimSmithAssignment1.docx, also the file must have a heading that includes your name, the assignment #, class number, date and 2 participants name).  

b) Upload the cb-ot-devst05.ad.wsu.edu URL of your webpage. We will attempt to run your project and compare it to other student submission to ensure academic integrity.

2. There are two components to each homework assignment, the first is required (max 8 points possible) the second is optional (max 2 points possible). The required portion asks you to complete and turn-in a program that covers the module topics and requested program functionality. Note turn-in means 2 files (.docx of pictures and code, URL of the webpage)

a) You can earn 1 more point (or fractional) by properly commenting your code (grading is professor’s or TA’s discretion). In general, place some commenting in your code every 3 lines to discuss the functionality that is occurring. These 'notes to self' make the program easier to maintain, should you (or worse a teammate) need to reopen it in the future. Commenting = documentation. Also space out the code so it is readable.

b) You can get up to 1 more point (or fractional) for adding more functionality to your program (such as incorporate a new control ,a new potion or methodology). Try to envision the person that would use this program to run their store. What more functionality or calculations would they want? You must clearly mention and highlight this original upgrade (grading is professor’s or TA’s discretion). Your extra work is appreciated and rewarded.

3. .5 point deduction for late work (less = than 3 days late). 1 point deduction for >3 days late and <= 2 weeks late. Work over 2 weeks late not accepted for grading. You can also gain .5 point for turning in the assignment more than 48 hours early.

4. If you worked with a partner on the assignment you must put their name on the submission. A request to add a second person's name to your submission after the submission is made, will not be accepted.