

BA272 Assignment 2

This assignment is designed to give you a chance to implement the programming techniques from Rob Miles' book from pages 32 through 66 plus pages 70 and 71. It includes the following main techniques which you should explore or even master:

- Constants
- The switch command
- Commenting source code
- Looping (three variations: Do While, and For)
- Methods (that share data)
- Nested loops
- Placeholders in Print Strings

Your coded result should require only a single .cs file (no other project files). Store that .cs file and the required test example text file in the `assign2\MyWork` folder. I recommend that you then try making a new project using that code and testing it. Also, please consider the grading criteria listed at the bottom of this page; your grade will be based on those criteria.

To get all that in, we need some fairly extensive directions. Here they are.

StapleMakers wants a program to project inventory levels for their one and only product (red staplers). Last year they produced 1100 staplers and sold 1050 and had 250 staplers on hand at the end of the year. They want to see how inventory will be affected if they experience different growth rates over several years. Production and sales are not the same each month. They close their production line in February and June to allow for vacations. Sales are also seasonal. StapleMakers wants to model their sales in this system as follows: Most months' sales are equal to 7% of annual sales but September (back-to-school) and December (Christmas) will each have 15%.

Inputs:

- Annual Sales Growth (as a percentage of the prior year) from -50% to +500%
- Annual Production Capacity Growth (as a percentage of the prior year) from -50% to +500%
- The number of years to project from 1 to 10

Output:

- Year
- Units produced per month and an annual total, Units sold per month, Inventory balance per month
- Annual Totals for Production and Sales

Project over 2 years, given:

Year	10% Production Growth Rate	10% Sales Growth Rate	250 units Beginning Inventory
Year 1	Production	Sales	End Inv.
Jan	121	81	290
Feb	0	81	209
Mar	121	81	249
Apr	121	81	290

May	121	81	330
Jun	0	81	249
Jul	121	81	289
Aug	121	81	329
Sep	121	173	277
Oct	121	81	317
Nov	121	81	357
Dec	121	173	305
Total	1210	1155	

Year	2	Production	Sales	End Inv.
Jan		133	89	349
Feb		0	89	260
Mar		133	89	304
Apr		133	89	349
May		133	89	393
Jun		0	89	304
Jul		133	89	348
Aug		133	89	392
Sep		133	191	335
Oct		133	89	379
Nov		133	89	423
Dec		133	191	366
Total		1331	1271	

And so on for however many years the user requests. You can format it differently if you like but be sure and review the grading criteria.

A couple of simplifying allowances:

- Hint: Using the Placeholder method for displaying values automatically takes care of rounding.
- Hint: Rounding differently can result in slightly different yet still acceptable answers. Just be sure you can confirm that you understand how yours were computed. The numbers shown here assume the annual production and sales figures should be “correct” while monthly figures and inventory balances are approximate and subject to rounding error.
- You can assume users will always type numbers that can be parsed into integers. If they don’t, it can blow up the program, no problem. We will deal with error handling later in the term. Again, if you want to get ahead and learn to use try/catch or tryparse, you may, and extra credit can apply. **DESPITE EXTRA CREDIT YOU CANNOT GET MORE THAN 100%!** The extra credit will only be allowed to offset deductions in your program.

Requirements:

- Briefly describe the input, output, and processing of your program in a comment block.
- Create a test example text file listing correct computations for at least two possible sets of inputs. You should calculate these results in a spread sheet or with a calculator so you are sure your program works correctly. Include this file with your code in the `assign2/MyWork` folder.
- Include at least one method in your program that computes a value (as a return, or by ref, or with an out variable) to accomplish a meaningful task. *Hint: perhaps a method for accepting values from the console might be good since you need to do that for three different values.*

- Store hardcoded input values in constants (pg. 49 in the text).
- Use the switch command to implement the seasonality rules.
- Use Placeholder type formatting somewhere in your program (p. 57-58 in the text).
- Use nested For loops to process each month for each year.
- Use Do While (with the test after processing as shown on p. 51) to accept input from the console.

Make sure you understand these issues in preparation for a quiz or exam:

- How is a Do While loop as used here, different from While and For loops?
- What is the difference between a constant and a regular variable? Can you think of a way that using a constant in this program affects program maintenance in this application?
- What are several ways a method might give back or change a value when it is called (understand return, ref, and out)?

Evaluation Criteria (out of 100):

- Documentation: (25)
 - o (10) Test information in the documentation file is included, listing at least 2 input sets and the corresponding expected results.
 - o (10) The comment block clearly describes input, output, and processing for the program.
 - o (5) Variable names indicate the type of variable and are meaningful.
- Components: (55) About half each for including the component and half if the component does what it should
 - o (10) A method is defined in the program to give back a value.
 - o (5) Several values from the instructions are stored in constants.
 - o (10) Nested For loops process each month within each year.
 - o (10) Do While looping is used to accept values from the console.
 - o (10) Placeholder formatting is used in the program.
 - o (10) The switch command implements the seasonality rules.
- Function: (20)
 - o (10) The program compiles and executes.
 - o (10) The program produces correct output for each month and year. Months, Years, and Totals should be labeled. (It can't if it doesn't compile!)
- Extra Credit: (10) (Maximum for the assignment is 100%!)
 - o (10) The program effectively uses try/catch or tryparse to handle invalid input.

Thought questions:

- Although this program explores some interesting programming techniques in a business context, it has a number of obvious shortcomings. List some of them. You can think a bit about the overly simple modeling done, but mostly consider how the inputs and parameters are gathered, how the results are presented, and what could go wrong.
- Although this assignment does not request that you use arrays, they could be helpful in this application. Can you see how?