

BA272 Assignment 1

Create your own program based on the program on page 24 of the C# Yellow Book.

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 `assign1\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.

Your program should perform some computation related to some business. The whole program should neatly fit in a couple of pages at the most. You may use one of the provided scenarios below or make up your own.

Requirements:

- Have your program display a brief description of the scenario (a few of sentences at most). Include a quotation mark and a new line character (p. 36) in your description. Cover the three parts of a problem specification as discussed in the text (p.16).
- Create a test example text file listing correct computations for several possible inputs.
- Accept data from the console:
 - o Accept at least one integer variable from the console (something you count which won't have decimal places).
 - o Accept at least one double value from the console (a value that will often need to include decimal places).
 - o Use at least one number "hard coded" in the program in a computation.
 - o "Prompt" the use for the values to be entered using `Console.Write()`.
- Display the results of computations using the accepted values.
 - o Perform at least two different computations: *add, subtract, multiply or divide*.
 - o Explicitly cast the integer into a double as part of a computation.
 - o Use `WriteLine()` to return your results.

Some possible scenarios: (make up details as needed)

- Your company paints the inside of houses. Given ceiling height and the number of feet of wall, how much paint is needed?
- Your company discounts sales. Given a number of items at a given price, compute the discount.
- Overhead costs are applied to each stapler produced in your factory based on a rate of 3.5 cents per labor hour. Given the number of staplers produced and the labor hours used to produce them, compute the overhead applied to each stapler.

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

- How is `Write()` different from `WriteLine()`?
- Did you need to cast the integer to a double in your computation? Why or why not?
- Why are special codes needed to display quotation marks?

Evaluation Criteria (out of 100):

- Documentation: (30)
 - o (10) Test information in the documentation file is included, listing at least 3 input sets and the corresponding expected results.
 - o (5) Does the displayed description include input, output, and how the data is processed?
 - o (5) The test narrative effectively uses escape sequences.
 - o (5) Variable names indicate the type of variable.
 - o (5) Variable names are meaningful.
- Inputs: (20)
 - o (5) The user is prompted about what to enter.
 - o (5) The display uses `.Write()`.
 - o (10) Both a double and an integer are accepted.
- Computations: (20)
 - o (10) Two different operators (+ - * /) were used.
 - o (10) An integer is cast to a double.
- Function: (30)
 - o (10) The program compiles and executes.
 - o (10) The program produces the expected results given the test inputs.
 - o (5) The results messages are meaningful.
 - o (5) The results are displayed using `WriteLine()`.

Thought questions:

- Could you compute the area of a circle with a given radius using the operators on page 20?
- What could go wrong with your program? What sorts of errors should we be concerned about?
- Of course this is a very simplistic exercise. What would make the application (your program) better?
- How good is your test plan? What important things are NOT addressed?

Ideas these questions might introduce:

- Good program documentation.
- Libraries of useful computations (modular design and reuse).
- Error handling.
- Comprehensive testing.
- Conditionals and graphical user interfaces.