Mr. Bellavia

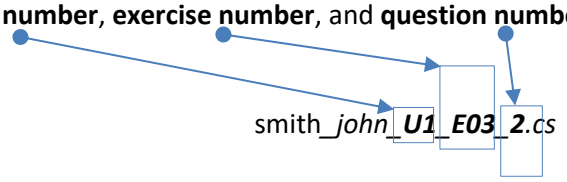
EXERCISE 05 - USER INPUT & BUILT-IN FUNCTIONS

IMPORTANT: Before submission, make a copy of your **'Program.cs'** file for each question and then rename each file to the following:

File Names:

- *last name_first name*_U1_E05_1.cs
- *last name_first name*_U1_E05_2.cs
- *last name_first name*_U1_E05_3.cs
- *last name_first name*_U1_E05_4.cs

Note: Along with last name and first name, make sure the end of the filename (i.e., before the **.cs**) has the **unit number**, **exercise number**, and **question number.** For example:



1. Write a program that will ask the user for **5 integer variables.** Have your program calculate the average of these variables and output the average (with 2 decimal places) to the **terminal** with an appropriate message. **Note:** This question is like question 2 from the last exercise, except now you are **asking the user** for the 5 integer variables.

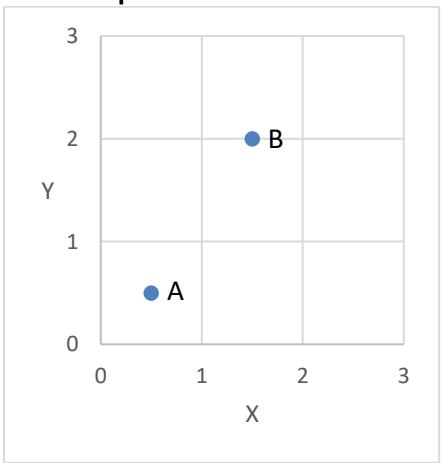
2. Write a program that asks the user for the number of apples, price per apple, and hst tax. Output the **subtotal** (number of apples * price per apple), **tax** (subtotal * tax percent), and **total** (subtotal + tax) to the terminal with appropriate messages.

   Note 1: Choose the appropriate **data types** for each input variable.

   Note 2: The 'hst tax' is a percentage inputted as a decimal. For example, for 13% the user is expected to type '.13'.

3. Write a program that asks the user for **two integers** ('a' & 'b') and then swap the values so that 'a' equals the value of 'b' and 'b' equals the value of 'a'. **Hint:** this will require the use of a third (temporary) variable.

4. Consider two points **A** & **B** on a **cartesian plane:**



These two points (**A** & **B**) both have an 'x' and 'y' value. We can represent these two points as follows:

A(x1, y1) & **B**(x2, y2)

The distance between two points can be calculated with the following formula:

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

Write a program that asks for the coordinates of points **A** & **B** (i.e., x1, y1, x2, & y2), then implement the above formula.

Note 1: Make sure your formula follows **BEDMAS,** avoid **truncation** and make use of the built-in **Math.Pow()** function.

Note 2: To calculate the square root you can use the built-in **Math.Sqrt()** function which takes only one parameter. Search 'C# square root' in google if you need more help.