Pair Programming 1 Tests

# Instructions

* **Always use the pair programming tests to ensure your program works properly.**
* **Take a screen shot with a white background of each execution in the tests.**
* **Only share with your partner work that you did together.**

1a. (5 points) Execute your program and compare its output to the Expected Output column. Unless otherwise noted, make your program output look like the Expected Output column.

|  |  |
| --- | --- |
| **Input [additional instructions in braces]** | **Expected Output** |
| [Ensure all variables are type int as in  int number1; ]  Enter 0 for the first number and 1 for the second | Enter first number: 0  Enter second number: 1  Sum: 1  Difference: -1  Product: 0  Quotient: 0 |
| Enter -7 for the first number and 2 for the second  [The quotient will just be -3 and not -3.5 because int variables can only hold integers and there is no rounding in the division operator.] | Enter first number: -7  Enter second number: 2  Sum: -5  Difference: -9  Product: -14  Quotient: -3 |
| [Change the program so all variables are type double as in double number1; then execute again. Now, the quotient will just be -3.5.]  Enter -7 for the first number and 2 for the second | Enter first number: -7  Enter second number: 2  Sum: -5  Difference: -9  Product: -14  Quotient: -3.5 |
| Enter 99.9 for the first number and 56.2 for the second  [If your program prints out a different number of digits to the right of the decimal, that’s okay.] | Enter first number: 99.9  Enter second number: 56.2  Sum: 156.1  Difference: 43.7  Product: 5614.38  Quotient: 1.77758 |
| [Change all variables back to type int. Since your variables are type int, the first cin statement reads the 99, but not the .9 and the second cin statement sees the .9 that hasn’t been read yet and that’s not an integer. So, the second cin statement does not read a value into the variable which probably has a 0 in it. Then, the program divides an integer by zero and crashes.]  Enter 99.9 for the first integer and 56.2 for the second | Enter first number: 99.9  Enter second number: Floating exception (core dumped) |

1b. (5 points) Execute your program and compare its output to the Expected Output column. Unless otherwise noted, make your program output look like the Expected Output column.

|  |  |
| --- | --- |
| **Input** | **Expected Output** |
| Enter 1 for mass and 1 for acceleration | Enter mass in kg: 1  Enter acceleration in m/s^2: 1  Force: 1 Newtons |
| Enter 1.1 for mass and 9.8 for acceleration | Enter mass in kg: 1.1  Enter acceleration in m/s^2: 9.8  Force: 10.78 Newtons |