

Caelan Osman

January 16, 2020

Lab 1 Report

Note: Hello World! Is output each time the code is run but is not involved in any of the below cases.

Case 1:

Because m, x, w, y, z are all floating-point data types, the operation will perform floating point arithmetic and print out and store an exact answer (up to floating point precision capabilities), the values will not be truncated.

Sample input:

1.2

1.4

1.5

1.8

Sample output:

Enter four numbers to average:

You entered: 1.2, 1.4, 1.5, 1.8

The mean is: 1.475

Case 2:

Note that I declared 4 more variables; a, b, c, d instead of casting w, x, y, z to integers. In this case, the right-hand side of the assignment will perform integer arithmetic, and if the sum of a, b, c, d is not divisible by 4, the answer stored in m will not be exact because the operation will truncate the value.

Sample input:

2

3

3

3

Sample output:

Enter four more numbers:

You entered: 2, 3, 3, 3

The mean is: 2

Note: If floating point arithmetic had been done the actual mean would be 2.75, the .75 got truncated. Also note, because a, b, c, d are integer types; you cannot enter a floating point.

Case 3:

a, b, c, d are all still integers with m float, but this time because the sum of a, b, c, d is being divided by a float namely 4.0, it will perform floating point division and store a floating point in m. However, because a, b, c, d are integer types, we can still not enter floating points into them. In general, either the denominator or numerator needs to be floating point to perform floating point arithmetic.

Sample Input:

Enter four final numbers:

2

3

3

3

Sample Output:

You entered: 2, 3, 3, 3

The mean is: 2.75

Note: even though we entered the same integers, because we divided by 4.0 we were given the exact mean of the four integers, namely 2.75.