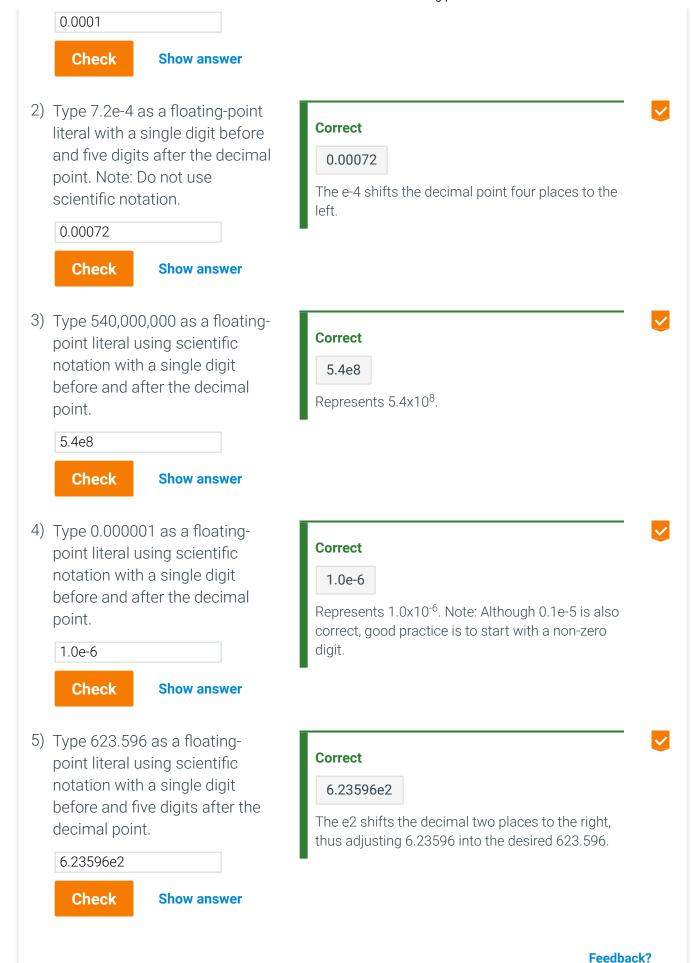
## 2.8 Scientific notation for floating-point literals

Scientific notation is useful for representing floating-point numbers that are much greater than or much less than 0, such as  $6.02 \times 10^{23}$ . A floating-point literal using **scientific notation** is written using an e preceding the power-of-10 exponent, as in 6.02e23 to represent  $6.02 \times 10^{23}$ . The e stands for exponent. Likewise, 0.001 is  $1 \times 10^{-3}$  and can be written as 1.0e-3. For a floating-point literal, good practice is to make the leading digit non-zero.

```
Figure 2.8.1: Calculating atoms of gold.
#include <iostream>
using namespace std;
int main() {
   double avogadrosNumber = 6.02e23; // Approximation of atoms per mole
   double gramsPerMoleGold = 196.9665;
   double gramsGold;
   double atomsGold;
   cout << "Enter grams of gold: ";</pre>
   cin >> gramsGold;
   atomsGold = gramsGold / gramsPerMoleGold * avogadrosNumber;
   cout << gramsGold << " grams of gold contains ";</pre>
   cout << atomsGold << " atoms" << endl;</pre>
   return 0;
Enter grams of gold: 4.5
4.5 grams of gold contains 1.37536e+22 atoms
                                                                                       Feedback?
PARTICIPATION
                2.8.1: Scientific notation.
ACTIVITY
1) Type 1.0e-4 as a floating-point
                                            Correct
   literal with a single digit before
   and four digits after the decimal
                                              0.0001
   point. Note: Do not use
                                            The e-4 shifts the decimal point four places to the
   scientific notation.
                                            left.
```



CHALLENGE ACTIVITY

2.8.1: Acceleration of gravity.



Compute the acceleration of gravity for a given distance from the earth's center, distCenter, assigning the result to accelGravity. The expression for the acceleration of gravity is:  $(G * M) / (d^2)$ , where G is the gravitational constant 6.673 x  $10^{-11}$ , M is the mass of the earth 5.98 x  $10^{24}$  (in kg) and d is the distance in meters from the earth's center (stored in variable distCenter).

```
1 #include <iostream>
2 using namespace std;
4 int main() {
      double G = 6.673e-11;
      double M = 5.98e24;
      double accelGravity;
      double distCenter;
 8
9
      cin >> distCenter;
10
11
      /* Your solution goes here */
12
      accelGravity = (G * M) / (distCenter*distCenter);
13
14
15
      cout << accelGravity << endl;</pre>
16
17
      return 0;
18 }
```

Run

✓ All tests passed

✓ Testing with distance 6.38e6 (earth's surface)

Your value 9.803495445209855

✓ Testing with distance 2\*6.38e6 (2x earth's radius)

Your value 2.4508738613024637

✓ Testing with distance 0.5\*6.38e6 (1/2 earth's radius)

Your value 39.21398178083942

✓ Testing with distance 1.000000 (at earth's core)

