

```
1 import components.simplereader.SimpleReader;
2 import components.simplereader.SimpleReader1L;
3 import components.simplewriter.SimpleWriter;
4 import components.simplewriter.SimpleWriter1L;
5
6 /**
7  * Program for calculating the square root of a number to within 0.01% relative
8  * error.
9  *
10 * @FayeLeigh
11 *
12 */
13 public final class Newton3 {
14
15     /**
16      * No argument constructor--private to prevent instantiation.
17      */
18     private Newton3() {
19     }
20
21     /**
22      * Checks if input is close to zero.
23      *
24      * @param x
25      *      number to be compared to number close to zero
26      * @return true if input is close enough to zero, false if input is not zero
27      */
28     private static boolean isZero(double x) {
29         final double eps = 1E-10;
30         return x < eps;
31     }
32
33     /**
34      * Computes estimate of square root of x to within relative error 0.01%.
35      *
36      * @param x
37      *      positive number to compute square root of
38      * @param error
39      *      positive number that sets error threshold of calculations
40      * @return estimate of square root
41      */
42     private static double sqrt(double x, double error) {
43         double r = x;
44         boolean flag = true;
45
46         if (isZero(x)) { //If input is zero, skip calculation and return 0
47             return 0.0;
48         }
49         while (flag) { //Compute square root of x until error is acceptable
50             r = (r + x / r) / 2; //Newton iteration formula
51             if (Math.abs(r * r - x) / x < error * error) { //Error calculation
52                 flag = false;
53             }
54         }
55         return r;
56     }
57
58     /**
59      * Main method.
```

```
60     *
61     * @param args
62     *         the command line arguments
63     */
64     public static void main(String[] args) {
65         SimpleReader in = new SimpleReader1L();
66         SimpleWriter out = new SimpleWriter1L();
67
68         final int digits = 4; //Number of decimal digits of output
69         boolean flag = true;
70         double input = 0.0, output = 0.0, error = 1.0;
71
72         //Prompt to ask if user wishes to continue
73         out.println("This program computes the square root "
74             + "of any positive number. ");
75         out.println("Would you like to continue? (y/n)");
76
77         //Sets flag to false if user does not enter "y"
78         String yn = in.nextLine();
79         if (!yn.equals("y")) {
80             flag = false;
81         }
82         if (flag) {
83             out.println("Please enter the desired error threshold as a "
84                 + "percentage (e.g. enter 0.01 for 0.01% error): ");
85             error = in.nextDouble() / 100.0; //Convert % to decimal and store
86         }
87
88         /*
89          * Until user declines, keep requesting numbers and outputting their
90          * square roots
91          */
92         while (flag) {
93             out.println("Enter any positive number: "); //Prompt for number
94             input = in.nextDouble();
95             output = sqrt(input, error); //Call method sqrt() to find sqrt of number
96             out.print("The square root of " + input + " is ");
97             out.print(output, digits, false);
98             out.println();
99             out.println("Would you like to enter another number? (y/n)");
100             yn = in.nextLine();
101             if (!yn.equals("y")) {
102                 flag = false;
103             }
104         }
105         in.close();
106         out.close();
107     }
108 }
```