```java
 1 import components.simplereader.SimpleReader;
 5
 6 /**
 7  * In this project, I ask the user if they want to calculate the
   square root of
 8  * a number and calculate the number within an error of 0.01.
   After some
 9  * calculations, the result is printed out excluding the number
   0.
10  *
11  * @author Mohamed Jama
12  *
13  */
14 public final class Newton2 {
15
16     /**
17      * No argument constructor--private to prevent
   instantiation.
18      */
19     private Newton2() {
20     }
21
22     /**
23      * Creates a final double number named ERROR_NUMBER which is
   set to 0.01 so
24      * that it deals with magic numbers and can not be changed
   since it is a
25      * final number. 0.01 is also the epsilon or the error
   estimate.
26      */
27     private static final double ERROR_NUMBER = 0.01;
28
29     /**
30      * Put a short phrase describing the static method Newton1
   here.
31      */
32     /**
33      * Computes estimate of square root of x to within relative
   error 0.01%.
```

```java
34          *
35          * @param x
36          *             positive number to compute square root of
37          * @return estimate of square root but if number is 0 the
   answer is
38          *             undefined.
39          */
40         private static double sqrt(double x) {
41             double r = x;
42             if (r != 0) {
43                 while (Math.abs(r * r - x) / x
44                         - (ERROR_NUMBER * ERROR_NUMBER) > 0) {
45                     r = (r + x / r) / 2;
46                 }
47             }
48             return r;
49
50         }
51
52         /**
53          * Main method.
54          *
55          * @param args
56          *             the command line arguments
57          */
58         public static void main(String[] args) {
59             SimpleReader in = new SimpleReader1L();
60             SimpleWriter out = new SimpleWriter1L();
61             double number;
62             String con;
63             out.println("Would you like to proceed and calculate?(y/
   n): ");
64             con = in.nextLine();
65             while (con.equals("y")) {
66                 out.println("Enter a positive double number: ");
67                 number = in.nextDouble();
68                 if ((number - 0) <= 0) {
69                     out.println("Undefined");
70                 }
```

```java
71              double result = sqrt(number);
72              if ((result - 0) > 0) {
73                  out.println("The square root of the number " +
   number
74                          + " within a relative error of 0.01 is "
   + result);
75                  out.println("Goodbye");
76              }
77          }
78          /*
79           * Close input and output streams
80           */
81
82          out.println("Goodbye");
83          in.close();
84          out.close();
85      }
86 }
87
```