```java
import components.simplereader.SimpleReader;

/**
 * In this project, I ask the user if they want to calculate the square root of
 * a number and calculate the number within an error of 0.01. After some
 * calculations, the result is printed out.
 *
 * @author Mohamed Jama
 *
 */
public final class Newton1 {

    /**
     * No argument constructor--private to prevent instantiation.
     */
    private Newton1() {
    }

    /**
     * Creates a final double number named ERROR_NUMBER which is set to 0.01 so
     * that it deals with magic numbers and can not be changed since it is a
     * final number. 0.01 is also the epsilon or the error estimate.
     */
    private static final double ERROR_NUMBER = 0.01;

    /**
     * Put a short phrase describing the static method Newton1 here.
     */
    /**
     * Computes estimate of square root of x to within relative error 0.01%.
     *
```

```
35      * @param x
36      *            positive number to compute square root of
37      * @return estimate of square root
38      */
39     private static double sqrt(double x) {
40         double r = x;
41         while (Math.abs(r * r - x) / x - (ERROR_NUMBER *
  ERROR_NUMBER) > 0) {
42             r = (r + x / r) / 2;
43         }
44         return r;
45     }
46 
47     /**
48      * Main method.
49      *
50      * @param args
51      *            the command line arguments
52      */
53     public static void main(String[] args) {
54         SimpleReader in = new SimpleReader1L();
55         SimpleWriter out = new SimpleWriter1L();
56         double number;
57         String con;
58         out.println("Would you like to proceed and calculate?(y/
  n): ");
59         con = in.nextLine();
60 
61         while (con.equals("y")) {
62             out.println("Enter a positive double number: ");
63             number = in.nextDouble();
64 
65             double result = sqrt(number);
66             out.println("The square root of the number " +
  number
67                     + " within a relative error of 0.01 is " +
  result);
68             out.println("Goodbye");
69         }
```

```
70          /*
71           * Close input and output streams
72           */
73
74          out.println("Goodbye");
75          in.close();
76          out.close();
77      }
78  }
79
```