```java
import components.simplereader.SimpleReader;
import components.simplereader.SimpleReader1L;
import components.simplewriter.SimpleWriter;
import components.simplewriter.SimpleWriter1L;

/**
 * A program that computes the square root of a number using Newton Iteration
 *
 * @author VishalKumar
 *
 */
public final class Newton1 {

    /**
     * Private constructor so this utility class cannot be instantiated.
     */
    private Newton1() {
    }

    /**
     * Computes estimate of square root of x to within relative error 0.01%.
     *
     * @param x
     *            positive number to compute square root of
     * @return estimate of square root
     */
    private static double sqrt(double x) {
        double r = x;

        while (!(((r * r) - x) / x < (.001 * .001))) {
            r = ((r + (x / r)) / 2);
        }
        return r;
    }

    /**
     * Main method.
     *
     * @param args
     *            the command line arguments
     */
    public static void main(String[] args) {
        SimpleReader in = new SimpleReader1L();
        SimpleWriter out = new SimpleWriter1L();

        // boolean to store whether or not user wants to do another round
        boolean another = false;

        // compute if user would like to calculate another square root
        out.print("Would You like to calculate a square root? (y/n) ");
        char answer = in.nextLine().charAt(0);

        if (answer == 'y') {
            another = !another;
        }

        // loop until user no longer wants to calculate a square root
```

```
58          while (another) {
59              out.print("Enter a positive double (#.##): ");
60              double num = in.nextDouble();
61              double rootNum = sqrt(num);
62              out.println("The square root of " + num + " is " + rootNum);
63
64              out.print(
65                      "Would You like to calculate another square root? (y/n): ");
66              answer = in.nextLine().charAt(0);
67
68              if (answer != 'y') {
69                  another = !another;
70              }
71
72          }
73          out.println("Goodbye!");
74          /*
75           * Close input and output streams
76           */
77          in.close();
78          out.close();
79      }
80
81  }
82
```