```
 1 import components.simplereader.SimpleReader;
 5
 6 /**
 7  * Put a short phrase describing the program here.
 8  *
 9  * @author Put your name here
10  *
11  */
12 public final class Newton1 {
13
14     /**
15      * No argument constructor--private to prevent
   instantiation.
16      */
17     private Newton1() {
18     }
19
20     /**
21      * Computes estimate of square root of x to within relative
   error 0.01%.
22      *
23      * @param x
24      *            positive number to compute square root of
25      * @return estimate of square root
26      */
27     private static double sqrt(double x) {
28         // initializing r which is the initial guess for x^(1/2)
29         double r = x;
30         double epsilon = 0.0001;
31         // changing r until its relative error is less than
   epsilon
32         while ((Math.abs((r * r) - x) / x) > (epsilon *
   epsilon)) {
33             r = (r + x / r) / 2.0;
34         }
35         return r;
36     }
37
38     /**
```

```
39      * Main method.
40      *
41      * @param args
42      *             the command line arguments
43      */
44     public static void main String[] args) {
45         SimpleReader in = new SimpleReader1L();
46         SimpleWriter out = new SimpleWriter1L();
47         /*
48          * Put your main program code here; it may call myMethod
  as shown
49          */
50
51         String choice = "y";
52         while (choice.charAt(0) == 'y') {
53             // asking the user to enter the number to be square
  rooted
54             out.print("Enter a positive number: ");
55             double num = in.nextDouble();
56             // making sure the number entered is positive
57             if (num < 0) {
58                 out.print("Enter a POSITIVE number: ");
59                 num = in.nextDouble();
60             }
61             // adding a second if statement so that there is no
  problem even if the user enters multiple negative numbers
62             if (num > 0) {
63                 double root = sqrt(num);
64                 out.println("Square root of the number is " +
  root);
65                 out.print("Would you like to find another square
  root? (y/n) ");
66                 choice = in.nextLine();
67             }
68
69         }
70         /*
71          * Close input and output streams
72          */
```

```
73          in.close();
74          out.close();
75      }
76
77  }
78
```

```java
import components.simplereader.SimpleReader;

/**
 * Put a short phrase describing the program here.
 *
 * @author Put your name here
 *
 */
public final class Newton2 {

    /**
     * No argument constructor--private to prevent instantiation.
     */
    private Newton2() {
    }

    /**
     * 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) {
        // initializing r which is the initial guess for x^(1/2)
        double r = x;
        if (r != 0) {
            double epsilon = 0.0001;
            // changing r until its relative error is less than epsilon
            while ((Math.abs((r * r) - x) / x) > (epsilon * epsilon)) {
                r = (r + x / r) / 2.0;
            }
        }
        return r;
    }
```

```
39
40      /**
41       * Main method.
42       *
43       * @param args
44       *             the command line arguments
45       */
46      public static void main String[] args) {
47          SimpleReader in = new SimpleReader1L();
48          SimpleWriter out = new SimpleWriter1L();
49          /*
50           * Put your main program code here; it may call myMethod
    as shown
51           */
52          String choice = "y";
53          while (choice.charAt(0) == 'y') {
54              // asking the user to enter the number to be square
    rooted
55              out.print("Enter a positive number: ");
56              double num = in.nextDouble();
57              // making sure the number entered is positive
58
59              if (num < 0) {
60                  out.print("Enter a POSITIVE number: ");
61                  num = in.nextDouble();
62              }
63              // adding a second if statement so that there is no
    problem even if the user enters multiple negative numbers
64
65              if (num >= 0) {
66                  double root = sqrt(num);
67                  out.println("Square root of the number is " +
    root);
68                  out.print("Would you like to find another square
    root? (y/n) ");
69                  choice = in.nextLine();
70              }
71
72          }
```

```
73          /*
74           * Close input and output streams
75           */
76          in.close();
77          out.close();
78      }
79
80  }
81
```

```java
import components.simplereader.SimpleReader;

/**
 * Put a short phrase describing the program here.
 *
 * @author Put your name here
 *
 */
public final class Newton3 {

    /**
     * No argument constructor--private to prevent
instantiation.
     */
    private Newton3() {
    }

    /**
     * Computes estimate of square root of x to within relative
error 0.01%.
     *
     * @param x
     *            positive number to compute square root of
     * @param epsilon
     *            value of epsilon
     * @return estimate of square root
     */
    private static double sqrt(double x, double epsilon) {
        // initializing r which is the initial guess for x^(1/2)

        double r = x;
        if (r != 0) {
            // changing r until its relative error is less than
epsilon

            while ((Math.abs((r * r) - x) / x) > (epsilon *
epsilon)) {
                r = (r + x / r) / 2.0;
            }
```

```
39          }
40          return r;
41      }
42
43      /**
44       * Main method.
45       *
46       * @param args
47       *             the command line arguments
48       */
49      public static void main String[] args) {
50          SimpleReader in = new SimpleReader1L();
51          SimpleWriter out = new SimpleWriter1L();
52          /*
53           * Put your main program code here; it may call myMethod
    as shown
54           */
55          // taking the value of epsilon as the input
56          out.print("Enter the value of epsilon: ");
57          double epsilon = in.nextDouble();
58          String choice = "y";
59          while (choice.charAt(0) == 'y') {
60              // asking the user to enter the number to be square
    rooted
61
62              out.print("Enter a positive number: ");
63              double num = in.nextDouble();
64              // making sure the number entered is positive
65
66              if (num < 0) {
67                  out.print("Enter a POSITIVE number: ");
68                  num = in.nextDouble();
69              }
70              // adding a second if statement so that there is no
    problem even if the user enters multiple negative numbers
71
72              if (num >= 0) {
73
74                  double root = sqrt(num, epsilon);
```

```
75                  out.println("Square root of the number is " +
    root);
76                  out.print("Would you like to find another square
    root? (y/n) ");
77                  choice = in.nextLine();
78              }
79
80          }
81          /*
82           * Close input and output streams
83           */
84          in.close();
85          out.close();
86      }
87
88  }
89
```

```java
import components.simplereader.SimpleReader;

/**
 * Put a short phrase describing the program here.
 *
 * @author Put your name here
 *
 */
public final class Newton4 {

    /**
     * No argument constructor--private to prevent
instantiation.
     */
    private Newton4() {
    }

    /**
     * Computes estimate of square root of x to within relative
error 0.01%.
     *
     * @param x
     *            positive number to compute square root of
     * @param epsilon
     *            value of epsilon
     * @return estimate of square root
     */
    private static double sqrt(double x, double epsilon) {
        // initializing r which is the initial guess for x^(1/2)

        double r = x;
        if (r != 0) {
            // changing r until its relative error is less than
epsilon

            while ((Math.abs((r * r) - x) / x) > (epsilon *
epsilon)) {
                r = (r + x / r) / 2.0;
            }
```

```
39          }
40          return r;
41      }
42
43      /**
44       * Main method.
45       *
46       * @param args
47       *             the command line arguments
48       */
49      public static void main String[] args) {
50          SimpleReader in = new SimpleReader1L();
51          SimpleWriter out = new SimpleWriter1L();
52          /*
53           * Put your main program code here; it may call myMethod
    as shown
54           */
55          // taking the value of epsilon as the input
56
57          out.print("Enter the value of epsilon: ");
58          double epsilon = in.nextDouble();
59          boolean choice = true;
60          while (choice) {
61              // asking the user to enter the number to be square
    rooted
62
63              out.print("Enter a number: ");
64              double num = in.nextDouble();
65              // ending the loop if the user enters a negative
    value
66
67              if (num < 0) {
68                  choice = false;
69              }
70              if (num >= 0) {
71                  double root = sqrt(num, epsilon);
72                  out.println("Square root of the number is " +
    root);
73              }
```

```
74          }
75          /*
76           * Close input and output streams
77           */
78          in.close();
79          out.close();
80      }
81
82  }
83
```