```
1 //Purpose: take input from a file and output the fraction value of the operation
2 //Caspian Peavyhouse
3 //CS101-02
4 //11/23/14
5
6 import java.util.*;
7 import java.io.*;
8
9 public class FractionMath
10 {
11 /*
12      Algorithm for main(args)
13 main(args)
14    File inputFile <-- new File(args[0])
15    Scanner input <-- new Scanner(inputFile)
16
17    File outputFile <-- new File(args[1])
18    FileWriter writerOutput <-- new FileWriter(outputFile)
19    String currentLine <-- new String(input.nextLine())
20
21    writerOutput.write("Fraction Math Version 1")
22    writerOutput.write("Written by Caspian Peavyhouse")
23    writerOutput.write("CS101-02")
24
25    while (input.hasNextLine())
26       String output <-- new String(checkLine(currentLine))
27       if (output.equals(""))
28          break;
29       writerOutput.write(output)
30       writerOutput.write("\n")
31
32       currentLine <-- input.nextLine()
33
34 */
35
36 /*
37      Data Table for main
38
39 Variable or Constant     Purpose
40 _____________________     ___________________
41 args                     parameter for main
42 output                   contains the String value to be written to the file
43 */
44
45    public static void main(String [] args)throws Exception
46    {
47       File inputFile = new File(args[0]);
48       Scanner input = new Scanner(inputFile);
49
50       File outputFile = new File(args[1]);
51       FileWriter writerOutput = new FileWriter(outputFile);
52       String currentLine = new String(input.nextLine());
53
54       writerOutput.write("FractionMath Version 1\n");
55       writerOutput.write("Written by Caspian Peavyhouse\n");
56       writerOutput.write("CS101-02\n\n");
57
58       while (input.hasNextLine())
59       {
60          String output = new String(checkLine(currentLine));
61          if (output.equals(""))
62          {
63             break;
64          }
65          writerOutput.write(output);
66          writerOutput.write("\n");
67
68
```

```
 69          currentLine = input.nextLine();
 70
 71       }
 72
 73       writerOutput.close();
 74
 75    }//main(args)
 76
 77
 78 /*
 79 checkLine(String currentLine)
 80
 81    String endSequence <-- new String("quit")
 82    String emptyString <-- new String("")
 83    if (currentLine.toLowerCase().contains(endSequence))
 84       return emptyString
 85    else
 86       return currentLine
 87 */
 88 /*
 89      Data Table for checkLine
 90
 91 Variable or Constant     Purpose
 92 _____________________     ___________________
 93 args                      parameter for checkLine
 94 endSequence              contains the String value "quit"
 95 emptyString              contains the empty string
 96 */
 97    private static String checkLine(String currentLine)
 98    {
 99       String endSequence = new String("quit");
100       String emptyString = new String("");
101       if (currentLine.toLowerCase().contains(endSequence))
102       {
103          return emptyString;
104       }
105       else
106       {
107          return currentLine;
108       }
109    }//checkLine
110
111    private void readAdd()
112    {
113       //Stub
114    }//readAdd
115
116    private void readSubtract()
117    {
118       //Stub
119    }//readSubtract
120
121    private void readMultiply()
122    {
123       //Stub
124    }//readMultiply
125
126    private void readDivide()
127    {
128       //Stub
129    }//readDivide
130
131    private void readReciprocal()
132    {
133       //Stub
134    }//readReciprocal
135
136 }//class Template
```