

```

1 import components.simplereader.SimpleReader;
2
3 /**
4  * Using user input to test the charming theory
5  *
6  * @author Put your name here
7  *
8  */
9
10 public final class ABCDGuesser3 {
11
12     /**
13      * Private constructor so this utility class cannot be instantiated.
14      */
15     private ABCDGuesser3() {
16
17     }
18
19     /**
20      * Tests the "charming theory" using user input
21      *
22      * @param m
23      *         the mathematical constant
24      * @param pnum
25      *         the array of personal numbers
26      */
27     private static double jagerFormula(double m, int[] pnum) {
28
29         double[] exponents = { 5.0, -4.0, -3.0, -2.0, -1.0, -0.5, -1.0, 3.0,
30                                0.25, 0, 5.0, 4.0, 3.0, 2.0, 1.0, 0.5, 1.0, 3.0, 0.25 };
31         double[][] outcomes = new double[5][17];
32
33         /*
34          * creates a 2d array of all possible outcomes for (each personal
35          * number)^exponents
36          */
37         for (int i = 0; i < pnum.length; i++) {
38             for (int j = 0; j < exponents.length; j++) {
39                 double pnumDouble = pnum[i];
40                 double outcome = Math.pow(pnumDouble, exponents[j]);
41                 outcomes[i][j] = outcome;
42             }
43         }
44
45         double error = 1; //approx - exact / exact
46         double counter = Math.pow(17.0, 5.0);
47         double bestEstimate = 932;
48         int[] index = new int[5];
49
50         while (error > 0.1 && counter > 0) {
51
52             for (int i = 0; i < exponents.length; i++) {
53                 for (int j = 0; j < exponents.length; j++) {
54                     for (int k = 0; k < exponents.length; k++) {
55                         for (int h = 0; h < exponents.length; h++) {
56                             for (int g = 0; g < exponents.length; g++) {
57                                 double sum = outcomes[0][i] * outcomes[1][j]
58                                     * outcomes[2][k] * outcomes[3][h]
59                                     * outcomes[4][g];

```

```

61         error = Math.abs(sum - m) / m;
62         counter--;
63         if (error < bestEstimate) {
64             bestEstimate = error;
65             index[0] = i;
66             index[1] = j;
67             index[2] = k;
68             index[3] = h;
69             index[4] = g;
70         }
71     }
72 }
73
74
75
76
77     System.out.println("(" + pnum[0] + "^" + exponents[index[0]] + ")( "
78         + pnum[1] + "^" + exponents[index[1]] + ")( " + pnum[2] + "^"
79         + exponents[index[2]] + ")( " + pnum[3] + "^"
80         + exponents[index[3]] + ")( " + pnum[4] + "^"
81         + exponents[index[4]] + "returns an approximation"
82         + " of your constant with an error of ");
83 }
84 return bestEstimate;
85 }
86
87 /**
88  * Main method.
89  *
90  * @param args
91  *     the command line arguments
92  */
93 public static void main(String[] args) {
94     SimpleReader in = new SimpleReader1L();
95     SimpleWriter out = new SimpleWriter1L();
96
97     /*
98      * create an array of user's favorite numbers and check for + and != 1
99      */
100
101     int[] pnum = new int[5]; //initializing variables
102     int count = 0;
103
104     System.out.println("Please enter 5 of your favorite integers"
105         + " (press enter after each): ");
106     while (count < 5) {
107
108         int user = in.nextInteger();
109         if (user != 1 && user > 0) {
110             pnum[count] = user;
111             count++;
112         } else {
113             System.out
114                 .println("Your numbers must be + and not equal to 1 :( "
115                     + "Please try again.");
116         }
117     }

```

```
118
119     /*
120     * ask user for mathematical constant and check for + and != 1
121     */
122     double m = 0;
123     System.out.println("Now enter a cool (positive, non-1) mathematical "
124         + "constant: ");
125     double userConstant = in.nextDouble();
126     if (userConstant != 1 && userConstant > 0) {
127         m = userConstant;
128     } else {
129         System.out.println("Your numbers must be + and not equal to 1. \n"
130             + "Please try again.");
131     }
132
133     System.out.print(100 * jagerFormula(m, pnum) + "%");
134
135     /*
136     * Close input and output streams
137     */
138     in.close();
139     out.close();
140 }
141 }
```