

```

1 package com.company;
2
3
4 import java.util.LinkedList;
5 import java.util.List;
6
7 public class ExceptionTest {
8     public static void main(String[] args) {
9         /*
10             testTryCatch1();
11             testTryCatch2();
12             int val = testTryCatch3();
13             System.out.println(val);
14             print(1000);
15         */
16     }
17
18
19
20     /**
21      * 1 -----
22      * Examine the method below for null pointer / null value aspect
23      * and mark lines or code sections you think it has risk.
24      */
25     private List<Character> doSomething(List<Character> charList) {
26
27         List<Character> returnList = null;
28         for (Character c:charList){
29             if (Character.isDigit(c)){
30                 if (returnList == null)
31                     returnList = new LinkedList<>();
32                 returnList.add(c);
33             }
34         }
35         return returnList;
36     }
37
38
39
40
41
42     /**
43      * 2 -----
44      * What will method testTryCatch1 print to console?
45      */
46     private static void testTryCatch1() {
47         try {
48             badMethodRntEx();
49             System.out.println("a");
50         } catch (RuntimeException e) {
51             System.out.println("b");
52         } catch (Exception e) {
53             System.out.println("c");
54         } finally {
55             System.out.println("d");
56         }
57         System.out.println("e");
58     }
59
60     private static void badMethodRntEx() {
61         throw new RuntimeException();
62     }
63

```

```

64  /**
65   * 3 -----
66   * What will method testTryCatch2 print to console?
67   */
68  private static void testTryCatch2() {
69      try {
70          badMethodError();
71          System.out.println("a");
72      } catch (Exception e) {
73          System.out.println("b");
74      } finally {
75          System.out.println("c");
76      }
77      System.out.println("d");
78  }
79
80  private static void badMethodError() {
81      throw new Error();
82  }
83
84
85
86
87
88  /**
89   * 4 -----
90   * What will method testTryCatch3 print to console?
91   */
92  private static int testTryCatch3() {
93      try {
94          badMethodException();
95          return 1;
96      } catch (RuntimeException e) {
97          System.out.println("RuntimeException occurred");
98          return 2;
99      } catch (Exception e) {
100         System.out.println("Exception occurred");
101         return 3;
102     } finally {
103         System.out.println("finally");
104     }
105 }
106
107 private static void badMethodException() throws Exception {
108     throw new Exception();
109 }
110
111
112
113
114  /**
115   * 5 -----
116   * What would be the output of print(1000)?
117   */
118  private static void print(int n) {
119      if (n > 4000)
120          return;
121      System.out.printf("%d\n", n);
122      print(2*n);
123      System.out.printf("%d\n", n);
124  }
125 }

```