No THROWS


```
/**
 * This program allows the user to trace program execution under different
 * scenarios involving exceptions. To use this program, make sure to download
 * (or create) the following RuntimeExceptions (all should be in the same folder
 * as this program): RedException BlueException YellowException GreenException
 * OrangeException
 *
 * To run the program, you must provide two command-line arguments: mmm ccc
 * where
 *   mmm represents the method and is c, d, or e
 * and
 *  ccc represents the color of the exception and is red, blue, yellow, green,
 *  or orange
 *
 * To use command-line arguments when you run this program using Eclipse:
 * 1) Right click on this source file in the "Package Explorer" window.
 * 2) Select "Run As" from the pop-up menu.
 * 3) Select "Run Configurations..." from the pop-up menu, which brings up the
 *    "Run Configurations" window.
 * 4) Click on the "(x)= Arguments" tab.
 * 5) Enter the arguments (separated by a space) in the "Program arguments:"
 *    text box.
 * 6) Click either the "Run" button or the "Apply" and "Run" buttons.
 *
 * @author Beck Hasti, copyright 2008-2014
 */
public class ExceptionTester {
        private static String color;  // holds the color from the command line
        private static String method; // holds the method from the command line


        public static void main(String[] args) {

                // Set up
                if (args.length != 2) {
                        method = "e"; color = "blue";
                        //method = ""; color = "";
                } else {
                   method = args[0];
                   color = args[1];
                   System.out.println("Args="+method+";"+color);}


                // This is where the interesting stuff starts to happen

                System.out.print("main[");

                try {
                        methodA();
                        System.out.print("after A,");

                        methodE();
                        System.out.print("after E,");

                } catch (RedException exc) {
                        System.out.print("red,");

                } catch (GreenException exc) {
                        System.out.print("green,");
                }

                System.out.println("]main");
        }

        /*
         * A private method with a try and one catch
         */
        private static void methodA() {
                System.out.print("A[");

                try {
                        methodB();
                        System.out.print("after B,");
```

COLOR = ""

METHOD = ""

(1) (9) (10) (11) (2) (7)

```
        } catch (BlueException exc) {
                System.out.print("blue,");
        }

        System.out.print("]A ");
}

/*
 * A private method with a try, two catches, and a throw
 */
private static void methodB() {
        System.out.print("B[");

        try {
                methodC();
                System.out.print("after C,");

        } catch (YellowException exc) {
                System.out.print("yellow,");
                throw new GreenException();

        } catch (RedException exc) {
                System.out.print("red,");
        }

        methodD();
        System.out.print("after D");

        System.out.print("]B ");
}

/*
 * A private exception generating method.
 */
private static void methodC() {
        if (method.equalsIgnoreCase("C")) {
                if (color.equalsIgnoreCase("red"))
                        throw new RedException();
                else if (color.equalsIgnoreCase("blue"))
                        throw new BlueException();
                else if (color.equalsIgnoreCase("green"))
                        throw new GreenException();
                else if (color.equalsIgnoreCase("yellow"))
                        throw new YellowException();
                else if (color.equalsIgnoreCase("orange"))
                        throw new OrangeException();
        }
}

/*
 * A private exception generating method.
 */
private static void methodD() {
        if (method.equalsIgnoreCase("D")) {
                if (color.equalsIgnoreCase("red"))
                        throw new RedException();
                else if (color.equalsIgnoreCase("blue"))
                        throw new BlueException();
                else if (color.equalsIgnoreCase("green"))
                        throw new GreenException();
                else if (color.equalsIgnoreCase("yellow"))
                        throw new YellowException();
                else if (color.equalsIgnoreCase("orange"))
                        throw new OrangeException();
        }
}

/*
 * A private exception generating method.
 */
private static void methodE() {
        if (method.equalsIgnoreCase("E")) {
```

NO THROWS


```
                        if (color.equalsIgnoreCase("red"))
                                throw new RedException();
                        else if (color.equalsIgnoreCase("blue"))
                                throw new BlueException();
                        else if (color.equalsIgnoreCase("green"))
                                throw new GreenException();
                        else if (color.equalsIgnoreCase("yellow"))
                                throw new YellowException();
                        else if (color.equalsIgnoreCase("orange"))
                                throw new OrangeException();
                }
        }
}
```

DOES NOTHING