



VERIFY PARAMETER VALUES

zyBook 4.10

VERIFYING PARAMETER VALUES

- We want to verify that parameter values meet **pre-conditions** for our method, where a pre-condition is a condition that **must be true before a method executes** in order to guarantee that the method can execute properly.
- If **pre-conditions are not met, we will throw an exception**, which will halt method execution and provide user with an error message.
- Throwing an exception:
 - Syntax Template **throw <exception>;**
 - **Exceptions are objects**, so we must construct them
throw new IllegalArgumentException();
 - Can include message to the user
throw new IllegalArgumentException("message");

AREA OF SQUARE EXAMPLE

Side length for a square must be a positive integer.

```
/**
 * Returns the area of a square with given side length
 *
 * @param side length of side of square
 * @return area of square with given side length
 * @throws IllegalArgumentException if non-positive side length
 */
public static int area(int side) {
    if (side <= 0) {
        throw new IllegalArgumentException("Non-positive side length: " + side);
    }
    return side * side;
}
```



```

1  /**
2   * The class computes the area of three squares
3   *
4   * @author Jessica Young Schmidt
5   */
6  public class AreaOfSquare {
7
8      /**
9       * Starts the program.
10      *
11      * @param args command line arguments
12      */
13     public static void main(String[] args) {
14         // Side length for squares
15         int sideA = 10;
16         int sideB = 5;
17         int sideC = 11;
18
19         // Area for squares
20         int areaA = area(sideA);
21         int areaB = area(sideB);
22         int areaC = area(sideC);
23
24         // Print area for squares
25         printArea("Square A", areaA);
26         printArea("Square B", areaB);
27         printArea("Square C", areaC);
28         System.out.println();
29
30         // Should throw exception
31         int areaD = area(-3);
32     }
33
34     /**
35      * Returns the area of a square with given side length
36      *
37      * @param side length of side of square
38      * @return area of square with given side length
39      * @throws IllegalArgumentException if non-positive side length
40      */
41     public static int area(int side) {
42         if (side <= 0) {
43             throw new IllegalArgumentException("Non-positive side length: " + side);
44         }
45         return side * side;
46     }
47
48     /**
49      * Prints the area of the given square
50      *
51      * @param name name of the square
52      * @param area area of the square
53      */
54     public static void printArea(String name, int area) {
55         System.out.println(name + ": Area = " + area + ".");
56     }
57
58 }

```

```
$ java -cp bin AreaOfSquare
```

```
Square A: Area = 100.
```

```
Square B: Area = 25.
```

```
Square C: Area = 121.
```

```
Exception in thread "main"
```

```
java.lang.IllegalArgumentException:
```

```
Non-positive side length: -3
```

```
at AreaOfSquare.area(AreaOfSquare.java:43)
```

```
at AreaOfSquare.main(AreaOfSquare.java:31)
```