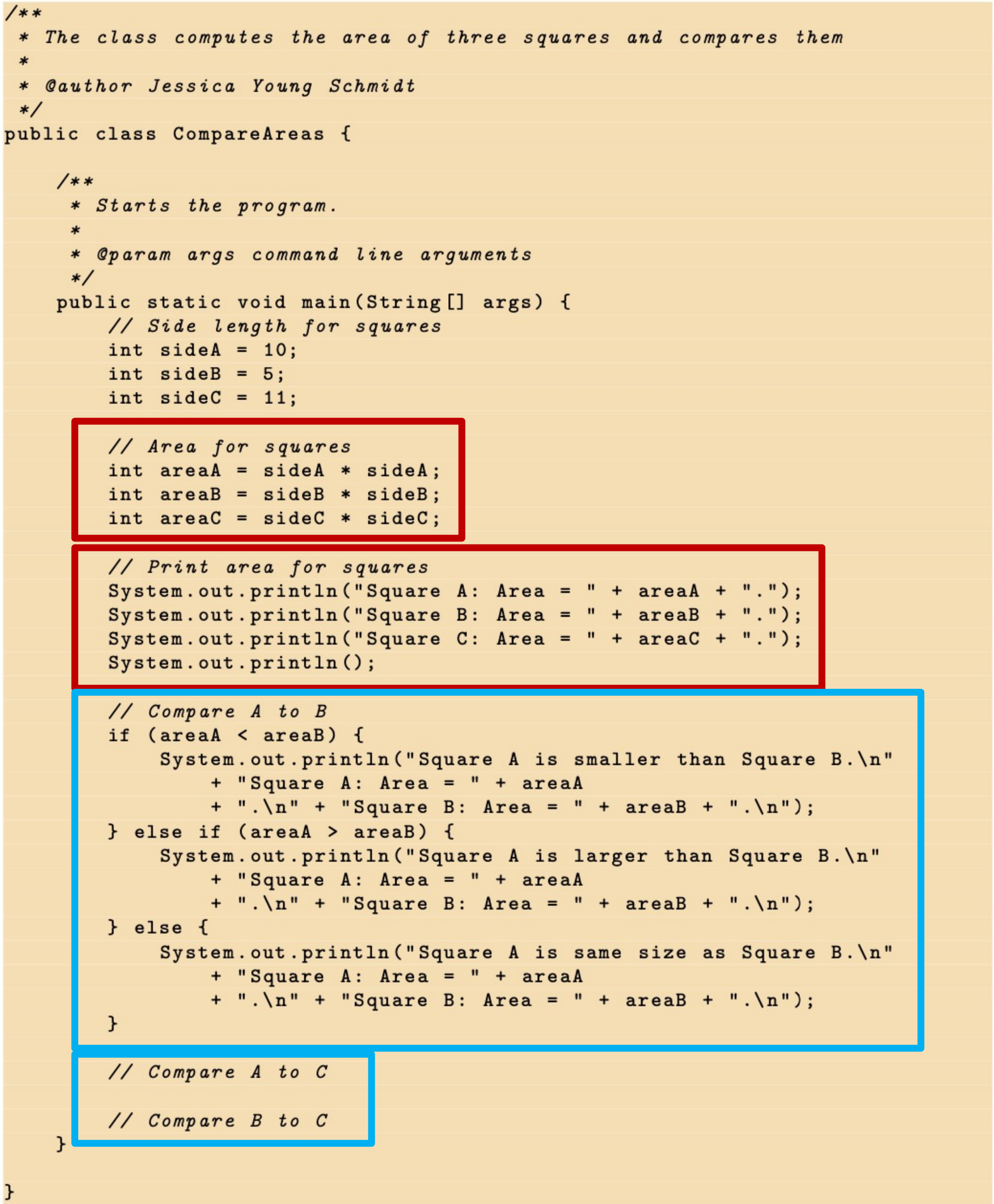
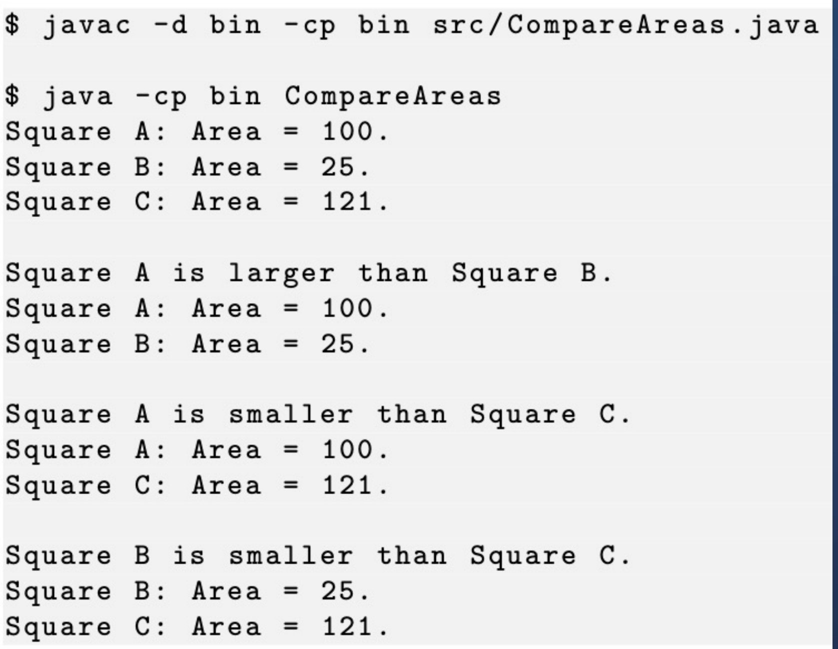
STATIC METHODS PARAMETERS AND RETURN VALUES

zyBook 4.1 – 4.8



```java
/**
 * The class computes the area of three squares and compares them
 *
 * @author Jessica Young Schmidt
 */
public class CompareAreas {

    /**
     * Starts the program.
     *
     * @param args command line arguments
     */
    public static void main(String[] args) {
        // Side length for squares
        int sideA = 10;
        int sideB = 5;
        int sideC = 11;

        // Area for squares
        int areaA = sideA * sideA;
        int areaB = sideB * sideB;
        int areaC = sideC * sideC;

        // Print area for squares
        System.out.println("Square A: Area = " + areaA + ".");
        System.out.println("Square B: Area = " + areaB + ".");
        System.out.println("Square C: Area = " + areaC + ".");
        System.out.println();

        // Compare A to B
        if (areaA < areaB) {
            System.out.println("Square A is smaller than Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        } else if (areaA > areaB) {
            System.out.println("Square A is larger than Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        } else {
            System.out.println("Square A is same size as Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        }

        // Compare A to C

        // Compare B to C
    }

}
```



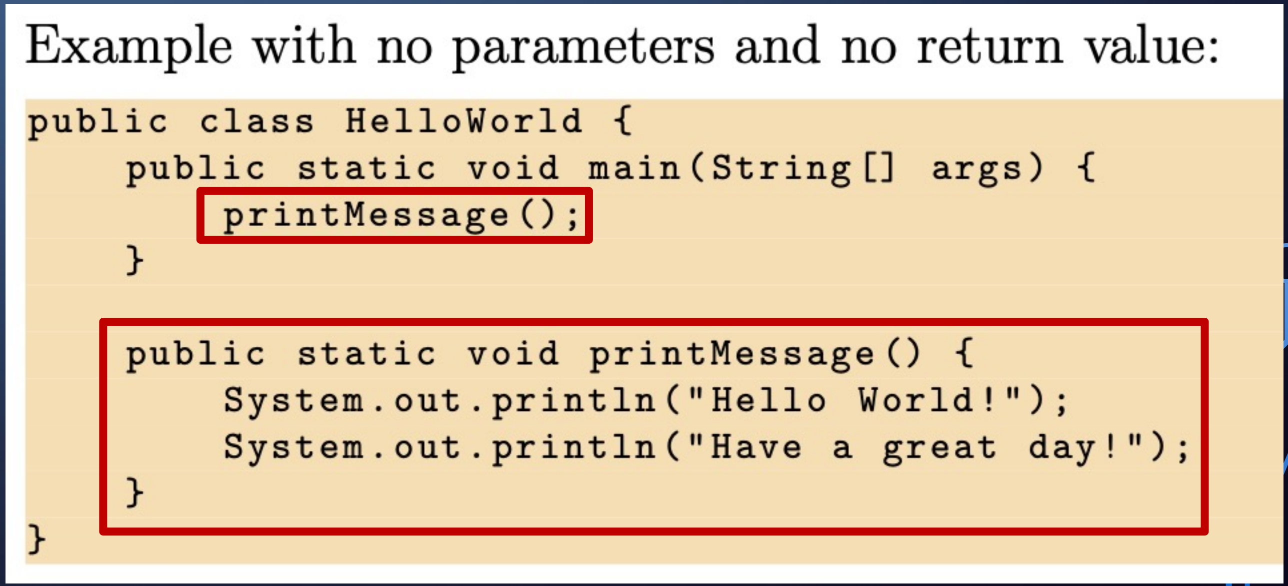
```
$ javac -d bin -cp bin src/CompareAreas.java

$ java -cp bin CompareAreas
Square A: Area = 100.
Square B: Area = 25.
Square C: Area = 121.

Square A is larger than Square B.
Square A: Area = 100.
Square B: Area = 25.

Square A is smaller than Square C.
Square A: Area = 100.
Square C: Area = 121.

Square B is smaller than Square C.
Square B: Area = 25.
Square C: Area = 121.
```

- Redundancy

- No structure

STATIC METHODS

- A static method is a group of statements with a given name.
  - Method declaration
    - Inside of the class, outside of the main
  - Method call
    - Inside of the main



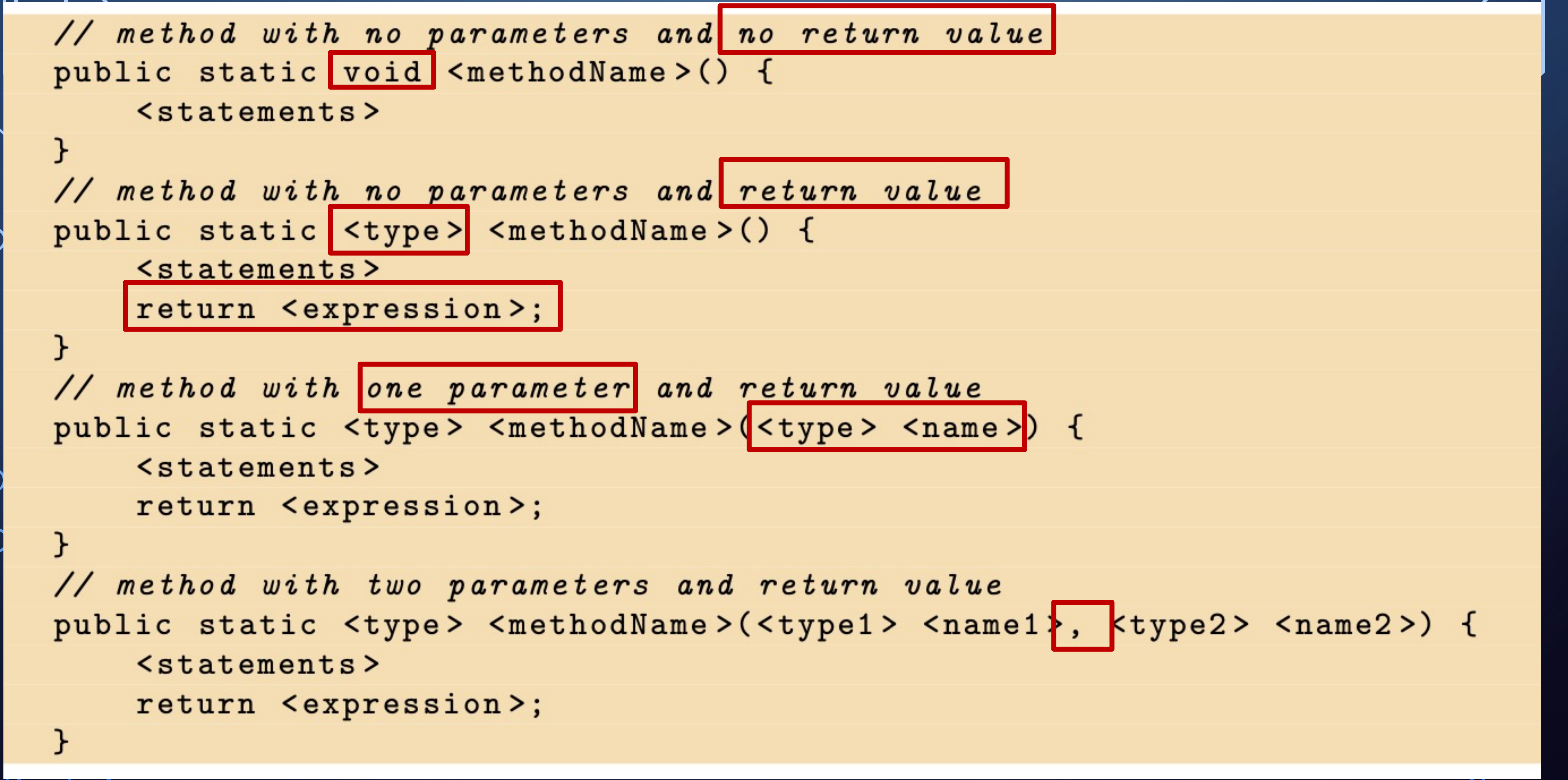
Example with no parameters and no return value:

```
public class HelloWorld {
    public static void main(String[] args) {
        printMessage();
    }

    public static void printMessage() {
        System.out.println("Hello World!");
        System.out.println("Have a great day!");
    }
}
```

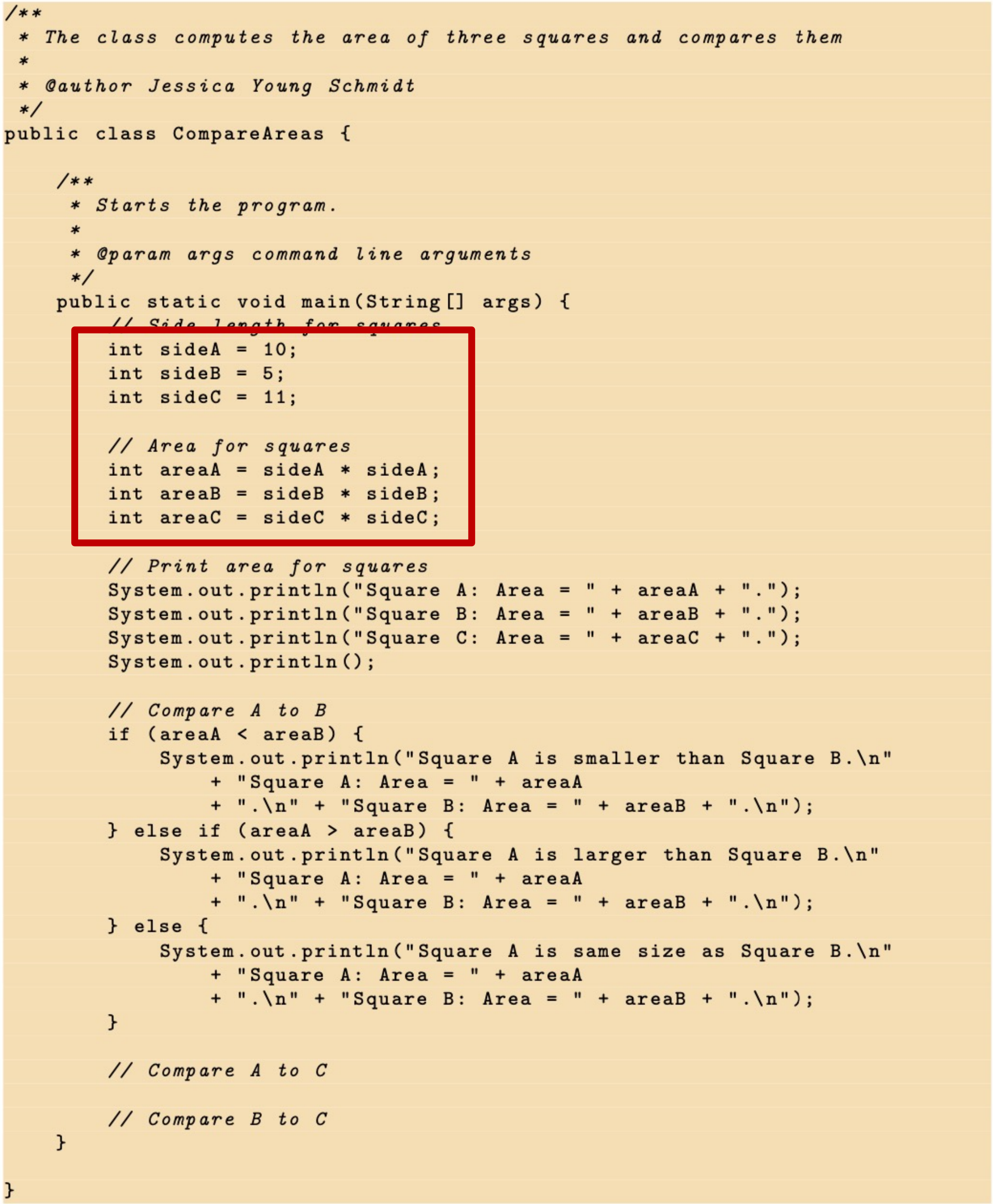
STATIC METHODS: PARAMETERS AND RETURN VALUE

- Parameters → Input into our method
  - Each method can have 0, 1, or many parameters.
  - Each parameter has a type and name (similar to variables).
  - In the method header, parameters are listed inside parentheses.
  - Scope of parameters is the method.
- Return Value → Output from our method
  - Each method can have a single return value.
  - Return type is listed in method header and must have return statement.

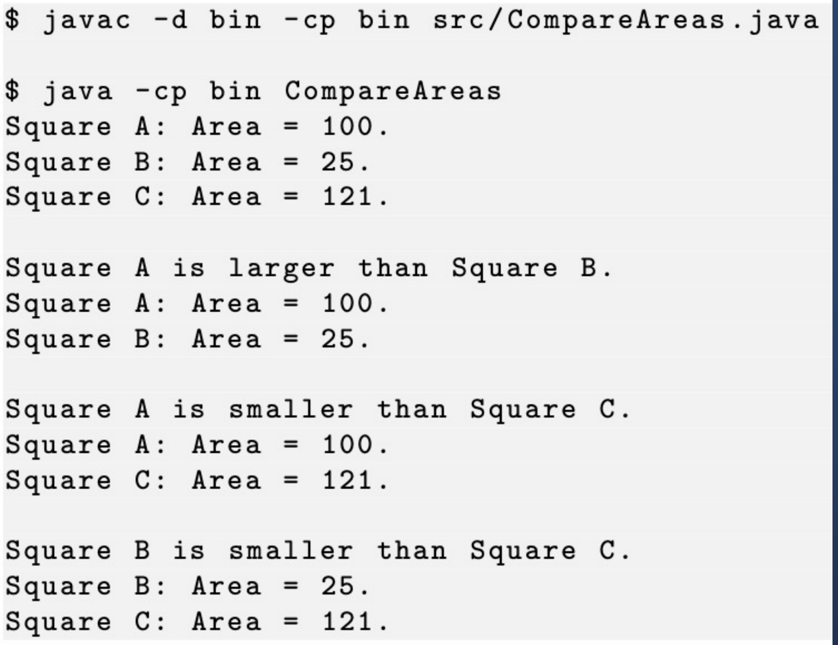
    If a method does not have a return value, we use void as return type.



```
// method with no parameters and no return value
public static void <methodName>() {
    <statements>
}
// method with no parameters and return value
public static <type> <methodName>() {
    <statements>
    return <expression>;
}
// method with one parameter and return value
public static <type> <methodName>(<type> <name>) {
    <statements>
    return <expression>;
}
// method with two parameters and return value
public static <type> <methodName>(<type1> <name1>, <type2> <name2>) {
    <statements>
    return <expression>;
}
```



```java
/**
 * The class computes the area of three squares and compares them
 *
 * @author Jessica Young Schmidt
 */
public class CompareAreas {

    /**
     * Starts the program.
     *
     * @param args command line arguments
     */
    public static void main(String[] args) {
        // Side length for squares
        int sideA = 10;
        int sideB = 5;
        int sideC = 11;

        // Area for squares
        int areaA = sideA * sideA;
        int areaB = sideB * sideB;
        int areaC = sideC * sideC;

        // Print area for squares
        System.out.println("Square A: Area = " + areaA + ".");
        System.out.println("Square B: Area = " + areaB + ".");
        System.out.println("Square C: Area = " + areaC + ".");
        System.out.println();

        // Compare A to B
        if (areaA < areaB) {
            System.out.println("Square A is smaller than Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        } else if (areaA > areaB) {
            System.out.println("Square A is larger than Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        } else {
            System.out.println("Square A is same size as Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        }

        // Compare A to C

        // Compare B to C
    }
}
```



```
$ javac -d bin -cp bin src/CompareAreas.java

$ java -cp bin CompareAreas
Square A: Area = 100.
Square B: Area = 25.
Square C: Area = 121.

Square A is larger than Square B.
Square A: Area = 100.
Square B: Area = 25.

Square A is smaller than Square C.
Square A: Area = 100.
Square C: Area = 121.

Square B is smaller than Square C.
Square B: Area = 25.
Square C: Area = 121.
```

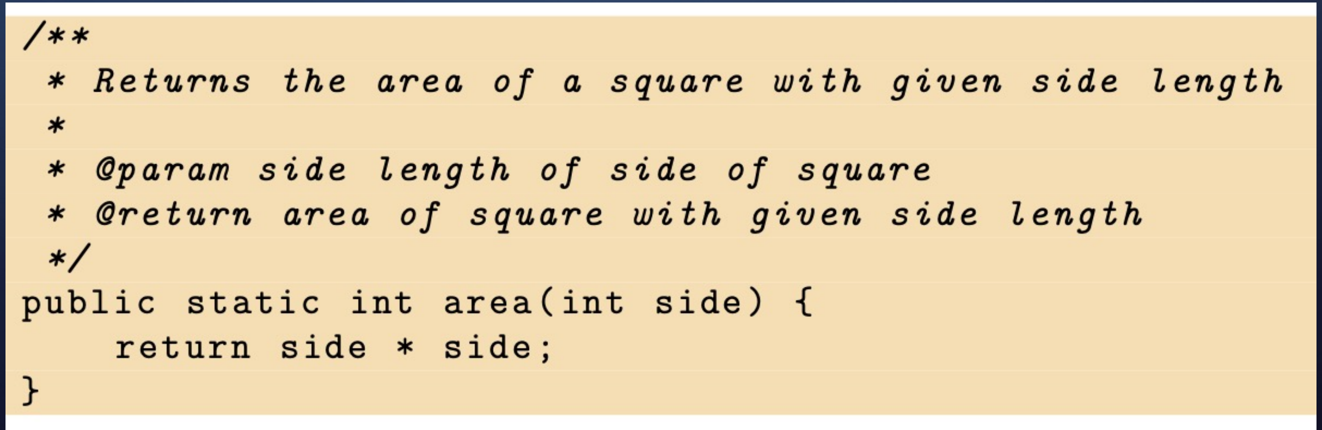
- Redundancy

- No structure

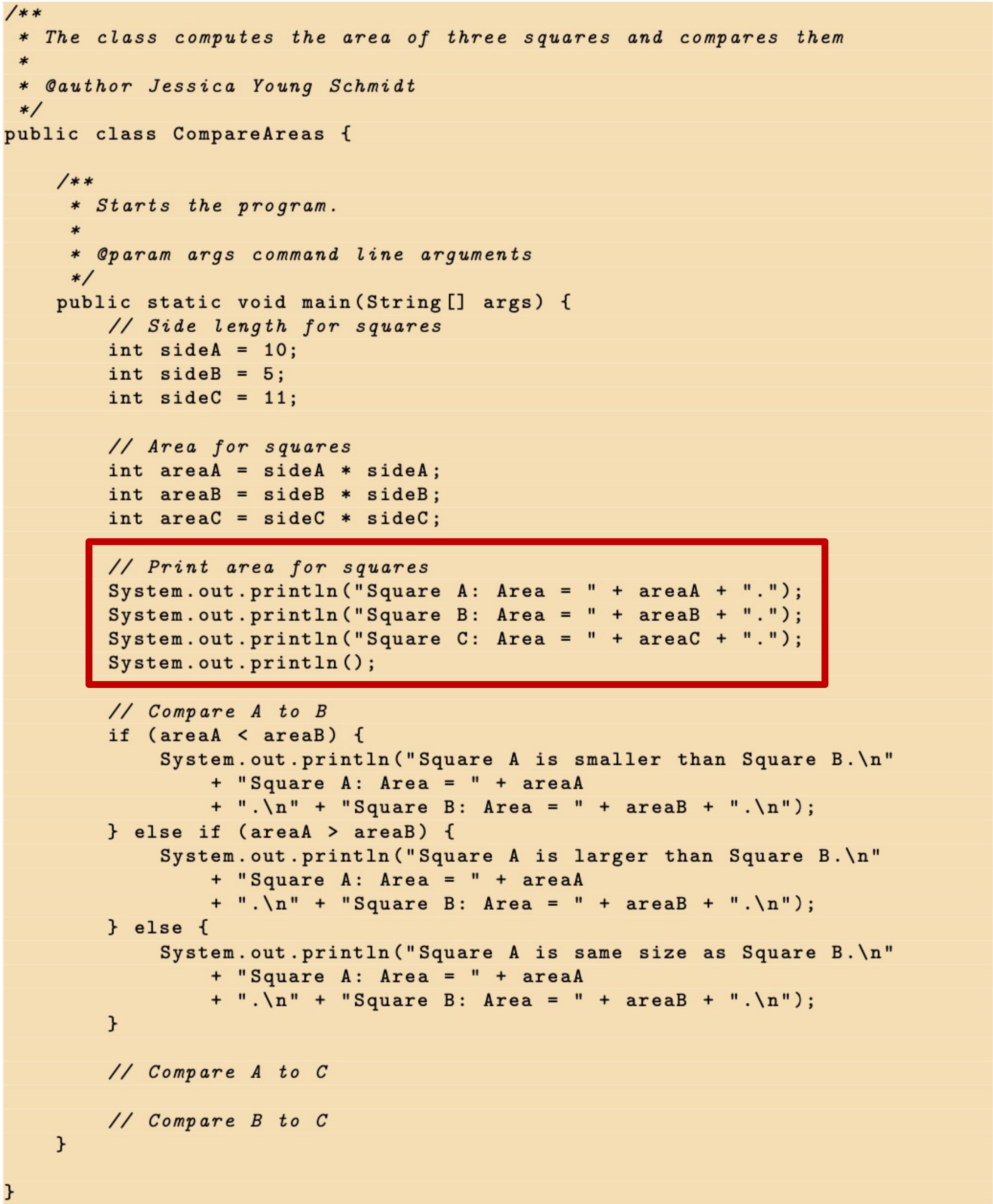
METHOD WITH PARAMETER AND RETURN VALUE

Example – area Method

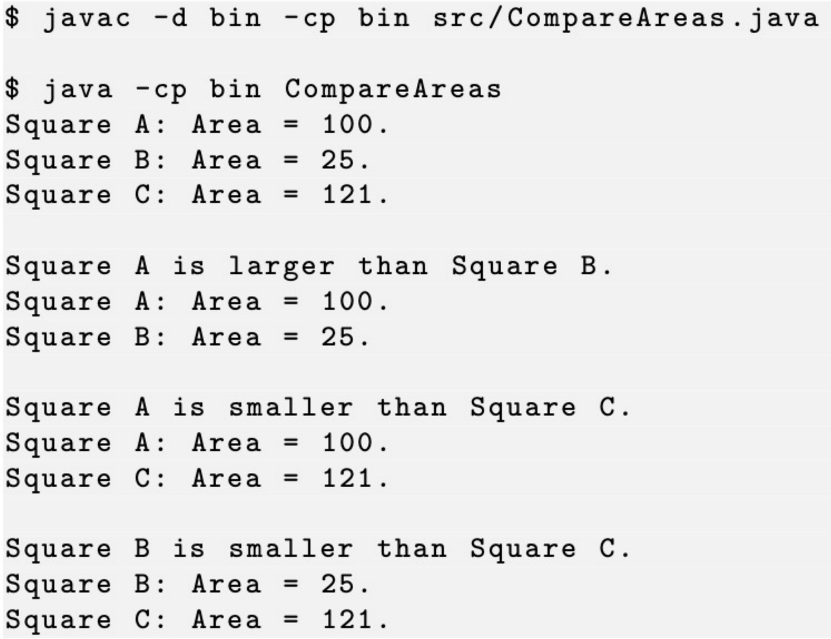
- Return type of int.
  - Since non-void return type, the method must have return statement.
- One parameter: int side.
  - int side is only accessible in area method and is not accessible in main method.
- When we call the method, we have to include value for parameter.
- We used the return value in assignment of variables.



```
/**
 * 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
 */
public static int area(int side) {
    return side * side;
}
```



```java
/**
 * The class computes the area of three squares and compares them
 *
 * @author Jessica Young Schmidt
 */
public class CompareAreas {

    /**
     * Starts the program.
     *
     * @param args command line arguments
     */
    public static void main(String[] args) {
        // Side length for squares
        int sideA = 10;
        int sideB = 5;
        int sideC = 11;

        // Area for squares
        int areaA = sideA * sideA;
        int areaB = sideB * sideB;
        int areaC = sideC * sideC;

        // Print area for squares
        System.out.println("Square A: Area = " + areaA + ".");
        System.out.println("Square B: Area = " + areaB + ".");
        System.out.println("Square C: Area = " + areaC + ".");
        System.out.println();

        // Compare A to B
        if (areaA < areaB) {
            System.out.println("Square A is smaller than Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        } else if (areaA > areaB) {
            System.out.println("Square A is larger than Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        } else {
            System.out.println("Square A is same size as Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        }

        // Compare A to C

        // Compare B to C
    }

}
```



```
$ javac -d bin -cp bin src/CompareAreas.java

$ java -cp bin CompareAreas
Square A: Area = 100.
Square B: Area = 25.
Square C: Area = 121.

Square A is larger than Square B.
Square A: Area = 100.
Square B: Area = 25.

Square A is smaller than Square C.
Square A: Area = 100.
Square C: Area = 121.

Square B is smaller than Square C.
Square B: Area = 25.
Square C: Area = 121.
```

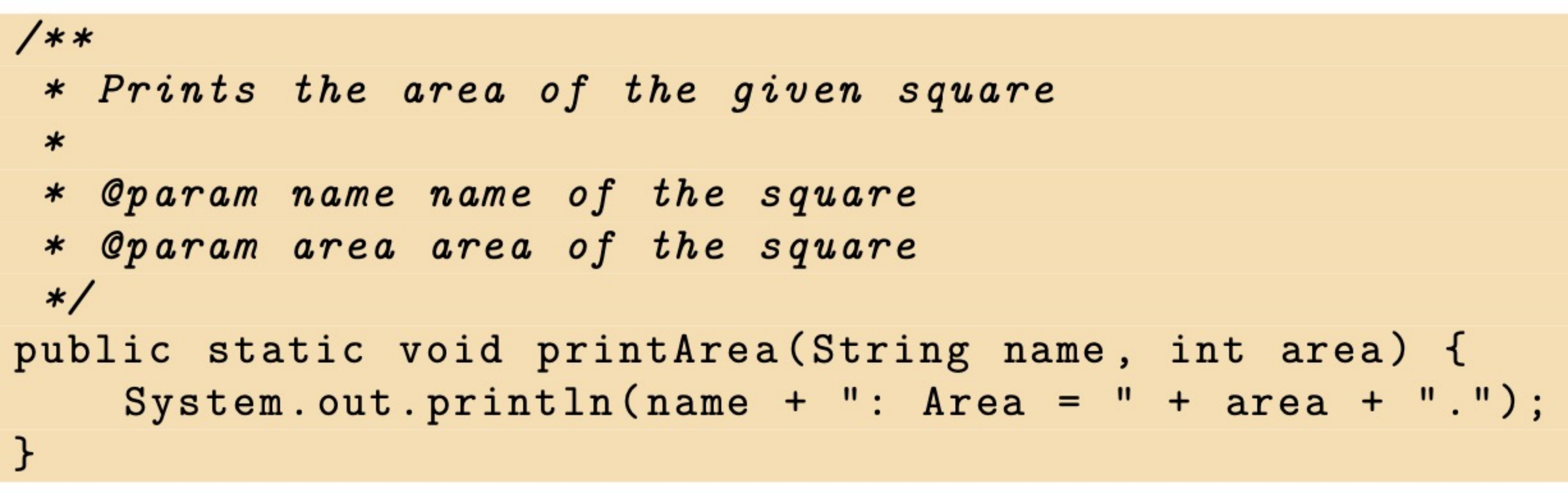
- Redundancy

- No structure

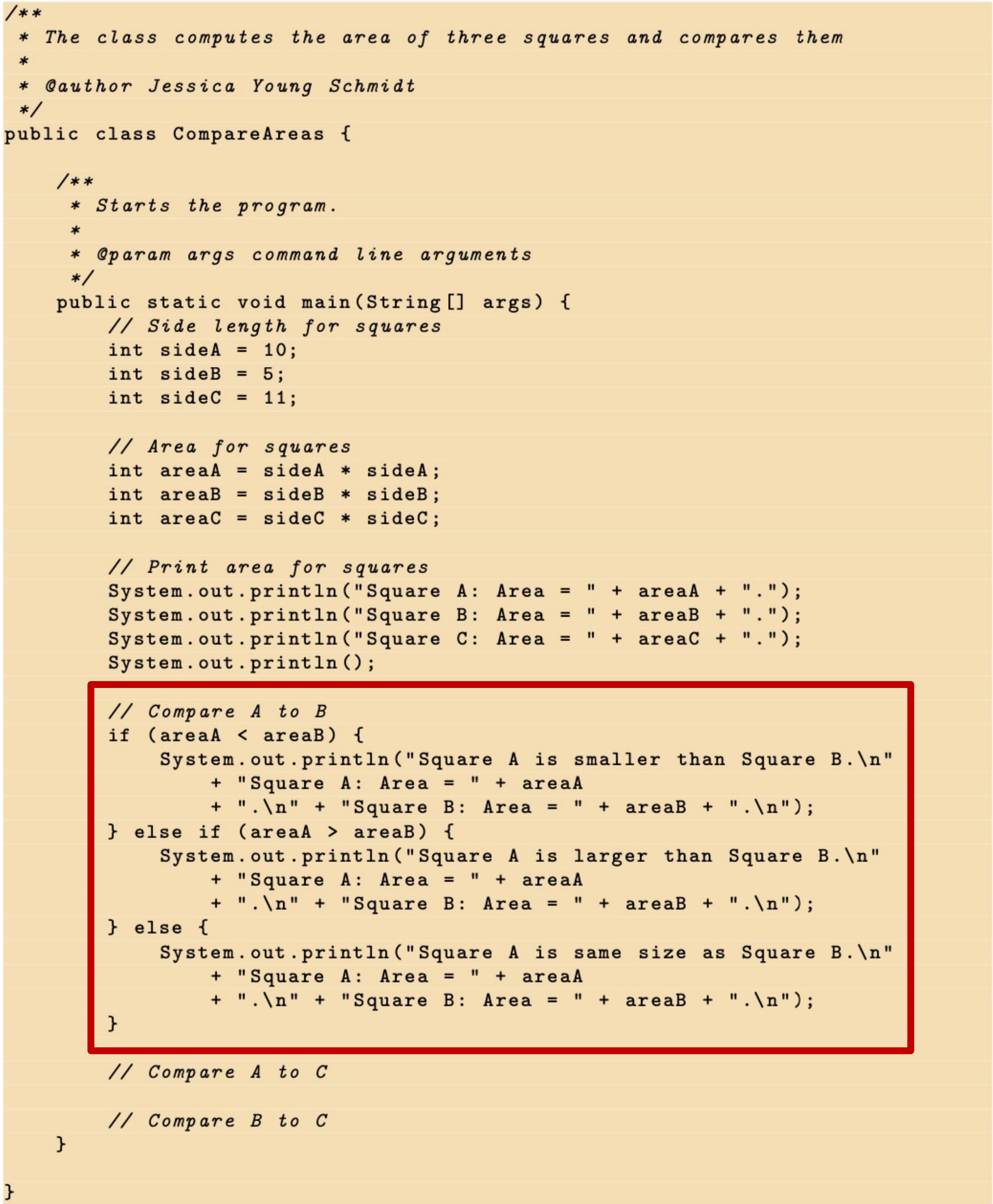
METHOD WITH PARAMETERS AND NO RETURN VALUE

Example – printArea Method

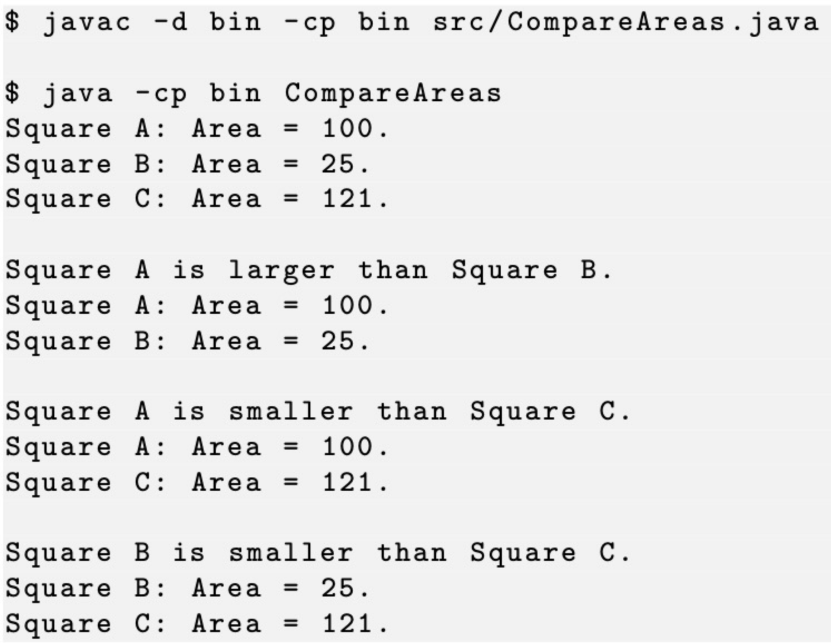
- No return type or value (void in method header)
- Two parameters: String name and int area
- When we call the method, we have to include values for each parameter.



```
/**
 * Prints the area of the given square
 *
 * @param name name of the square
 * @param area area of the square
 */
public static void printArea(String name, int area) {
    System.out.println(name + ": Area = " + area + ".");
}
```



```java
/**
 * The class computes the area of three squares and compares them
 *
 * @author Jessica Young Schmidt
 */
public class CompareAreas {

    /**
     * Starts the program.
     *
     * @param args command line arguments
     */
    public static void main(String[] args) {
        // Side length for squares
        int sideA = 10;
        int sideB = 5;
        int sideC = 11;

        // Area for squares
        int areaA = sideA * sideA;
        int areaB = sideB * sideB;
        int areaC = sideC * sideC;

        // Print area for squares
        System.out.println("Square A: Area = " + areaA + ".");
        System.out.println("Square B: Area = " + areaB + ".");
        System.out.println("Square C: Area = " + areaC + ".");
        System.out.println();

        // Compare A to B
        if (areaA < areaB) {
            System.out.println("Square A is smaller than Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        } else if (areaA > areaB) {
            System.out.println("Square A is larger than Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        } else {
            System.out.println("Square A is same size as Square B.\n"
                + "Square A: Area = " + areaA
                + ".\n" + "Square B: Area = " + areaB + ".\n");
        }

        // Compare A to C

        // Compare B to C
    }

}
```



```
$ javac -d bin -cp bin src/CompareAreas.java

$ java -cp bin CompareAreas
Square A: Area = 100.
Square B: Area = 25.
Square C: Area = 121.

Square A is larger than Square B.
Square A: Area = 100.
Square B: Area = 25.

Square A is smaller than Square C.
Square A: Area = 100.
Square C: Area = 121.

Square B is smaller than Square C.
Square B: Area = 25.
Square C: Area = 121.
```

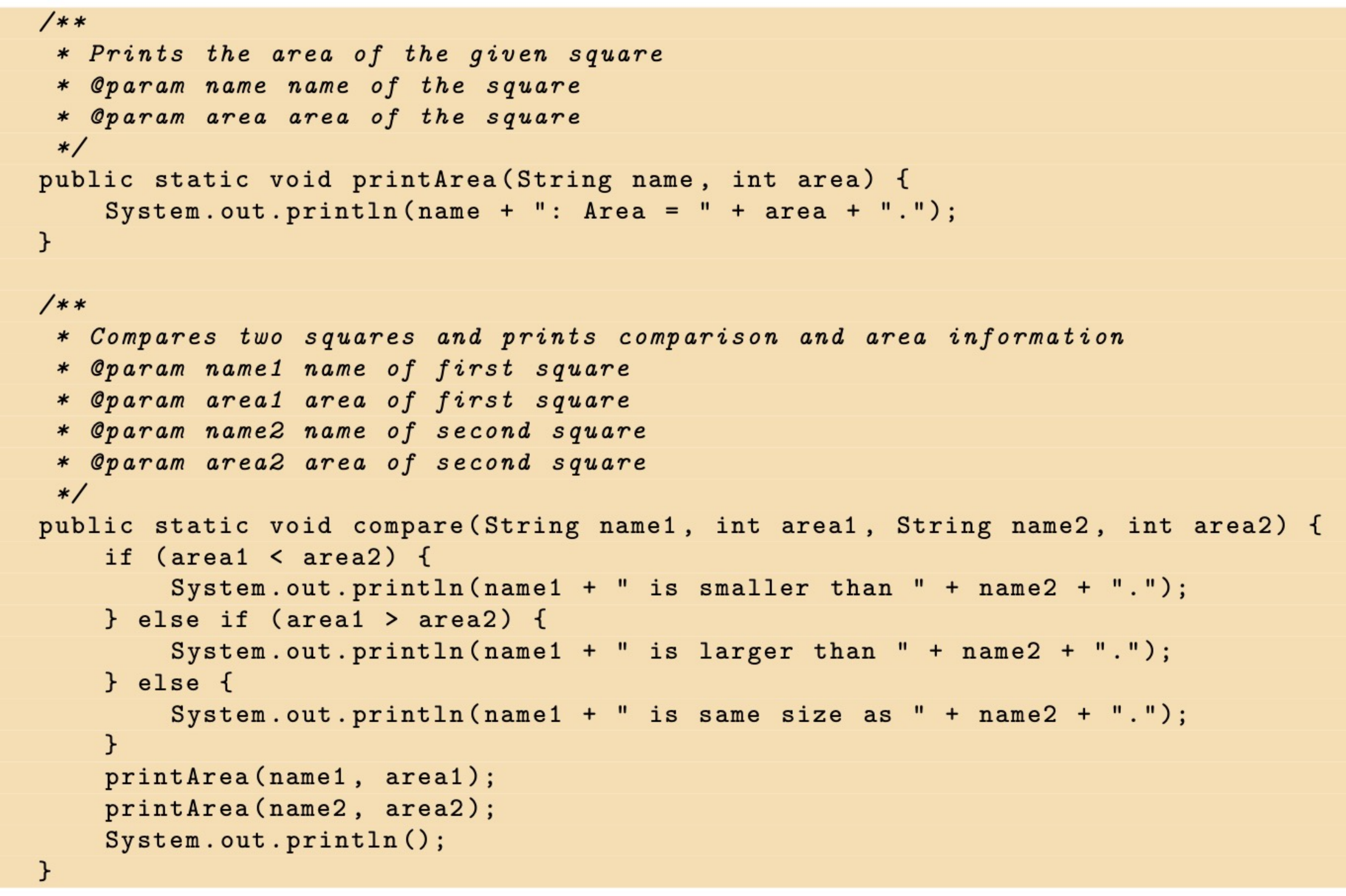
- Redundancy

- No structure

CALLING ANOTHER METHOD

compare Method

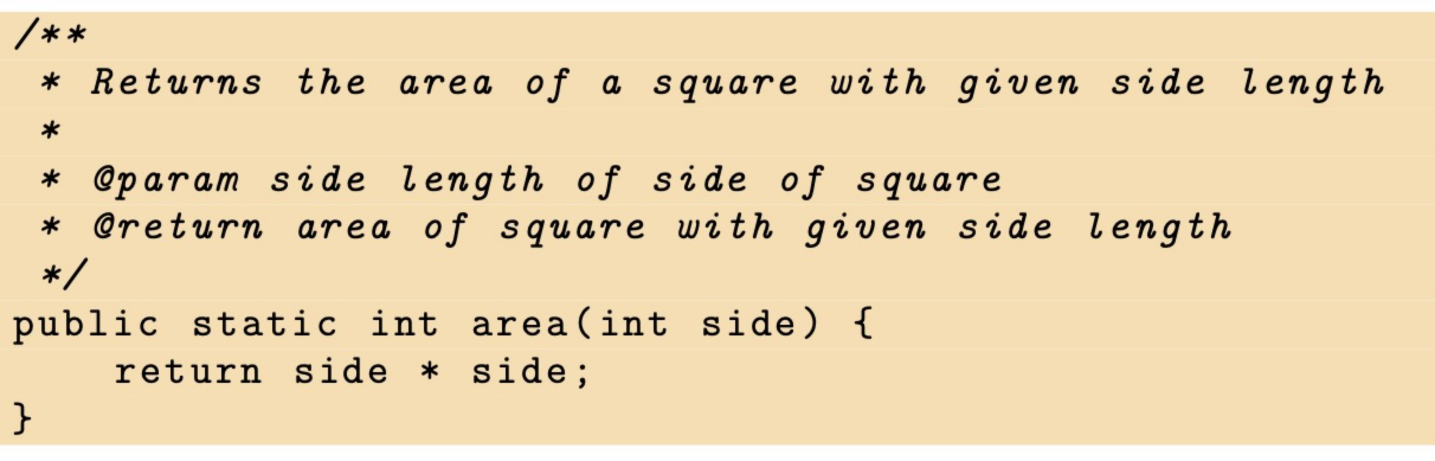
- No return value.
- Parameters for name and area of each square.
- Calls printArea method.



```java
/**
 * Prints the area of the given square
 * @param name name of the square
 * @param area area of the square
 */
public static void printArea(String name, int area) {
    System.out.println(name + ": Area = " + area + ".");
}

/**
 * Compares two squares and prints comparison and area information
 * @param name1 name of first square
 * @param area1 area of first square
 * @param name2 name of second square
 * @param area2 area of second square
 */
public static void compare(String name1, int area1, String name2, int area2) {
    if (area1 < area2) {
        System.out.println(name1 + " is smaller than " + name2 + ".");
    } else if (area1 > area2) {
        System.out.println(name1 + " is larger than " + name2 + ".");
    } else {
        System.out.println(name1 + " is same size as " + name2 + ".");
    }
    printArea(name1, area1);
    printArea(name2, area2);
    System.out.println();
}
```

METHOD JAVADOC

- Description of method
- @param tag for each parameter. List parameter name followed by description of parameter.
- If return value, @return tag followed by description of return.



```
/**
 * 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
 */
public static int area(int side) {
     return side * side;
}
```