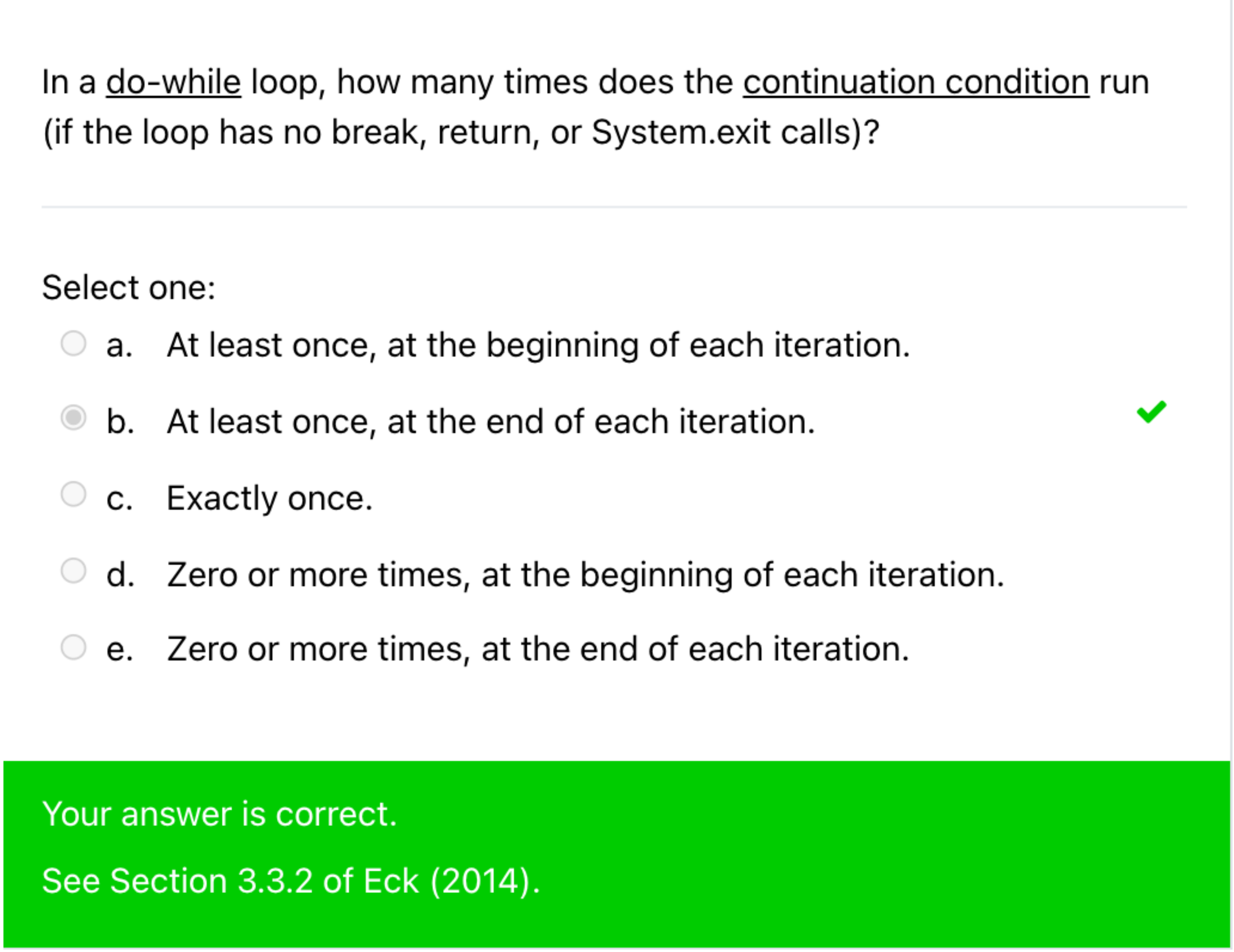
University of the People
Practice Final All Units
Course: CS 1102

In a do-while loop, how many times does the continuation condition run (if the loop has no break, return, or System.exit calls)?



In a do-while loop, how many times does the continuation condition run (if the loop has no break, return, or System.exit calls)?

Select one:

○ a. At least once, at the beginning of each iteration.

● b. At least once, at the end of each iteration. ✓

○ c. Exactly once.

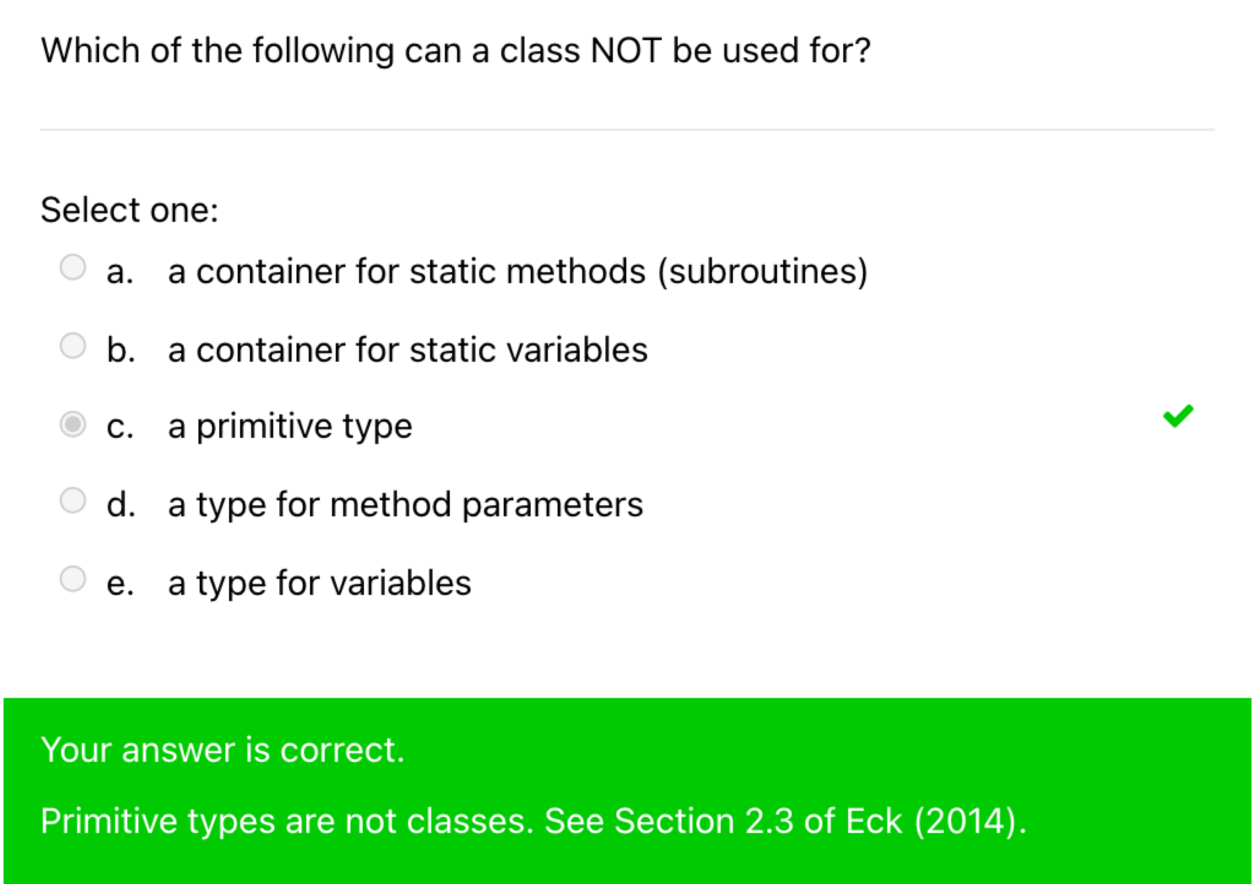
○ d. Zero or more times, at the beginning of each iteration.

○ e. Zero or more times, at the end of each iteration.

Your answer is correct.

See Section 3.3.2 of Eck (2014).

Which of the following can a class NOT be used for?



Which of the following can a class NOT be used for?

Select one:

○ a. a container for static methods (subroutines)

○ b. a container for static variables

● c. a primitive type ✓

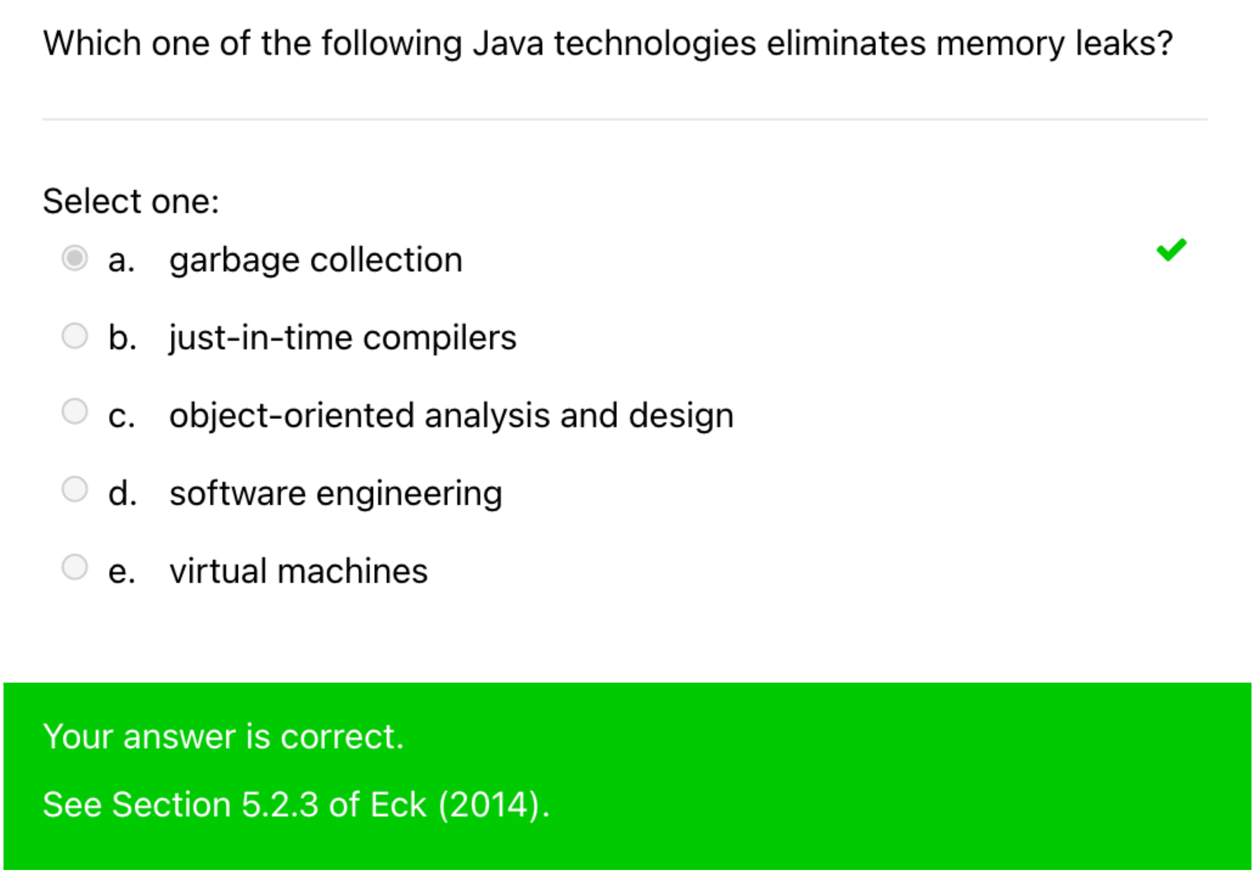
○ d. a type for method parameters

○ e. a type for variables

Your answer is correct.

Primitive types are not classes. See Section 2.3 of Eck (2014).

Which one of the following Java technologies eliminates memory leaks?



Which one of the following Java technologies eliminates memory leaks?

Select one:

● a. garbage collection ✓

○ b. just-in-time compilers

○ c. object-oriented analysis and design

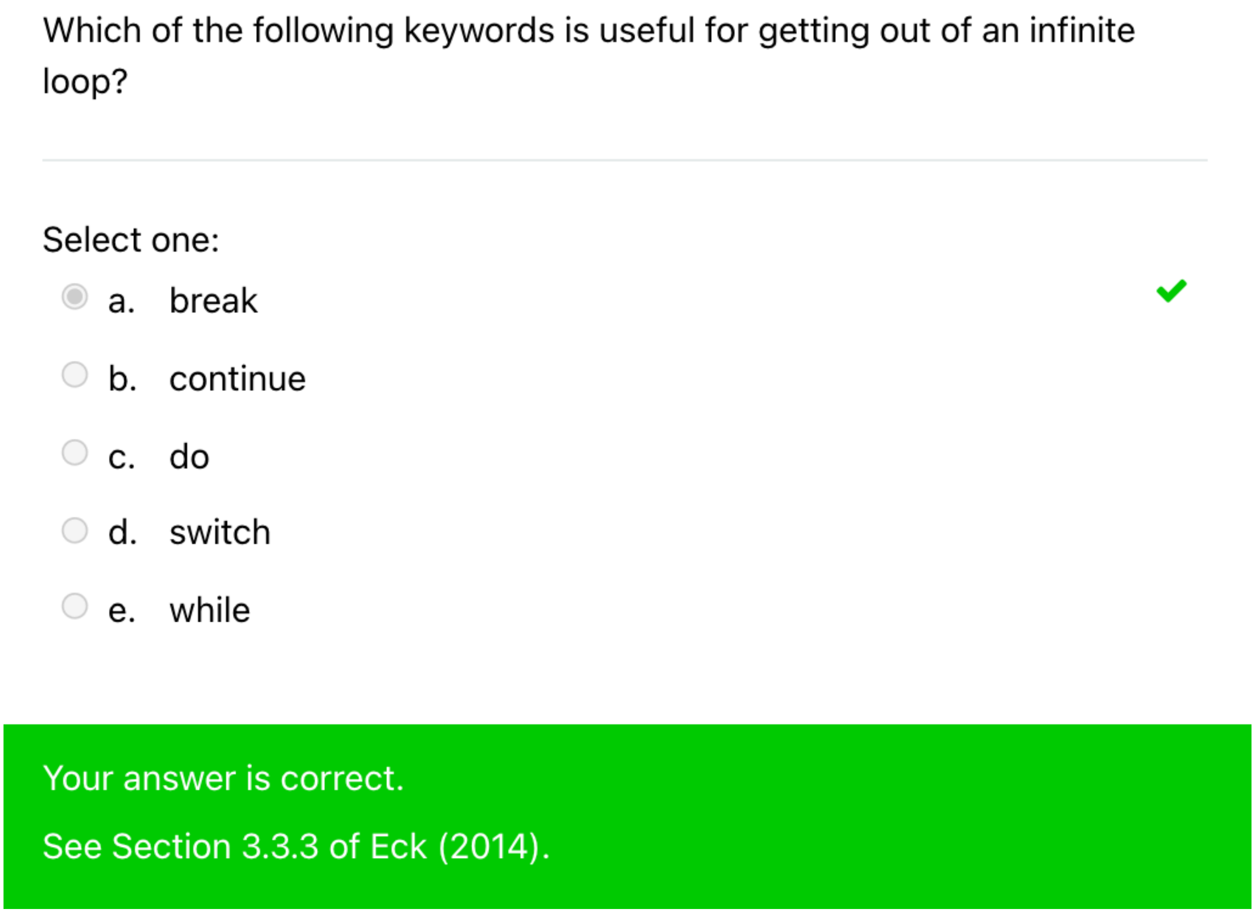
○ d. software engineering

○ e. virtual machines

Your answer is correct.

See Section 5.2.3 of Eck (2014).

Which of the following keywords is useful for getting out of an infinite loop?



Which of the following keywords is useful for getting out of an infinite loop?

Select one:

● a. break ✓

○ b. continue

○ c. do

○ d. switch

○ e. while

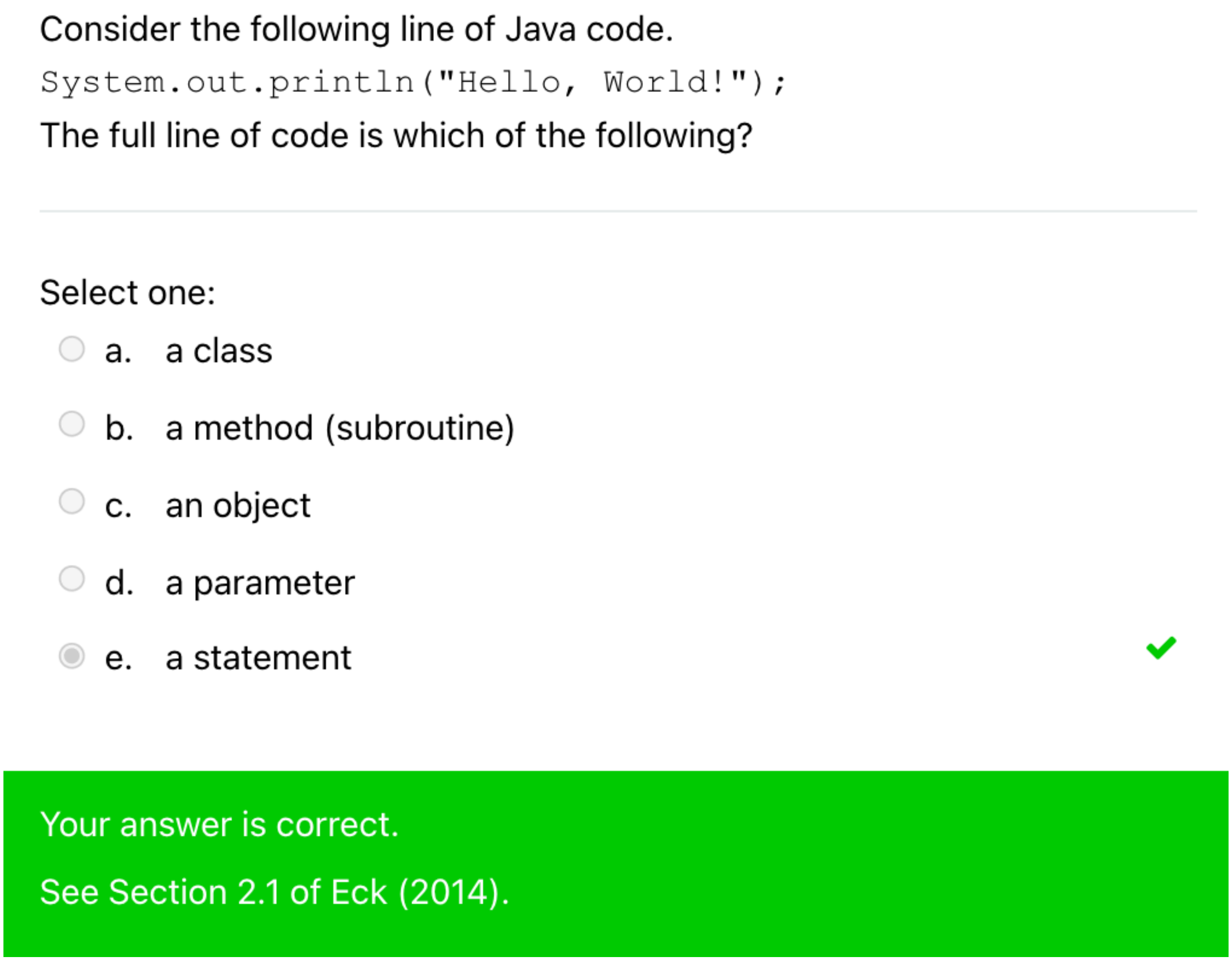
Your answer is correct.

See Section 3.3.3 of Eck (2014).

Consider the following line of Java code.

```
System.out.println("Hello, World!");
```

The full line of code is which of the following?



Consider the following line of Java code.

```
System.out.println("Hello, World!");
```

The full line of code is which of the following?

Select one:

○ a. a class

○ b. a method (subroutine)

○ c. an object

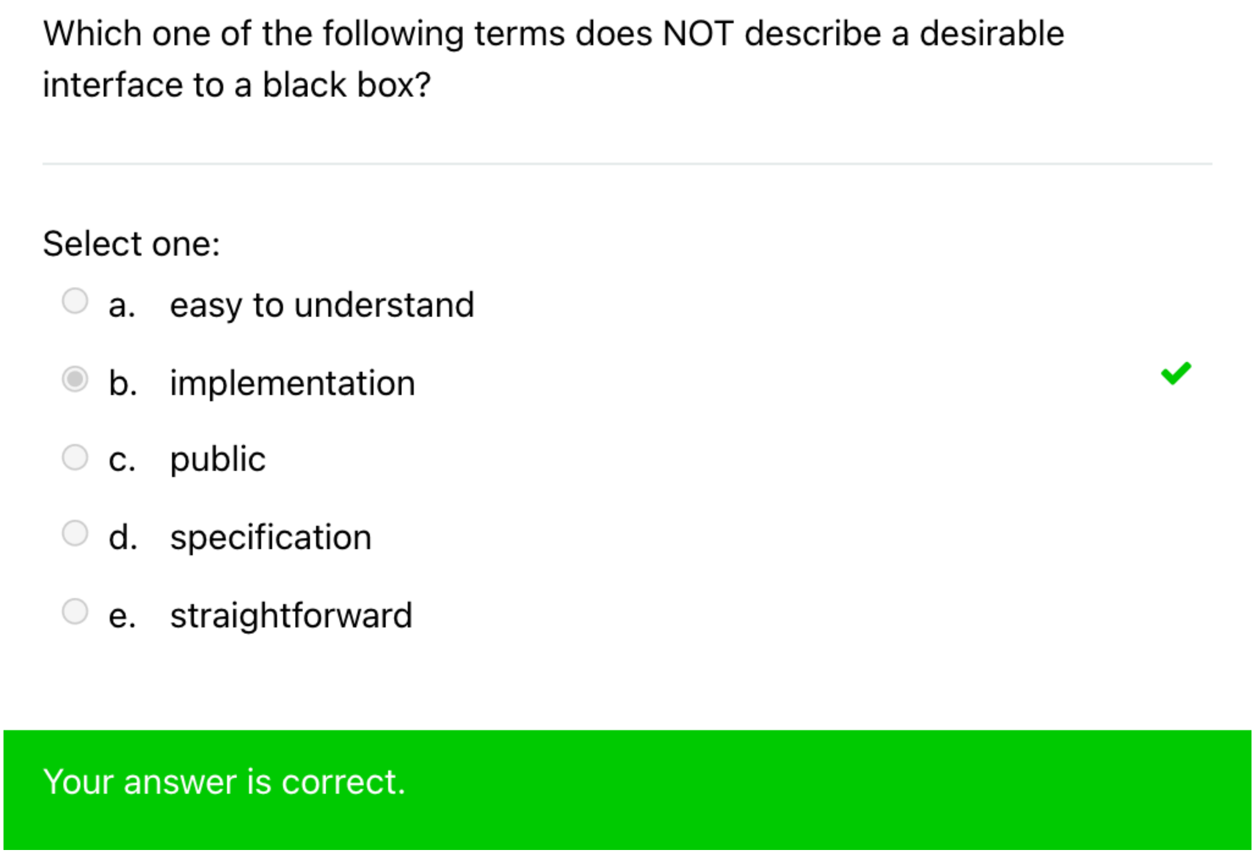
○ d. a parameter

● e. a statement ✓

Your answer is correct.

See Section 2.1 of Eck (2014).

Which one of the following terms does NOT describe a desirable interface to a black box?



Which one of the following terms does NOT describe a desirable interface to a black box?

Select one:

○ a. easy to understand

● b. implementation ✓

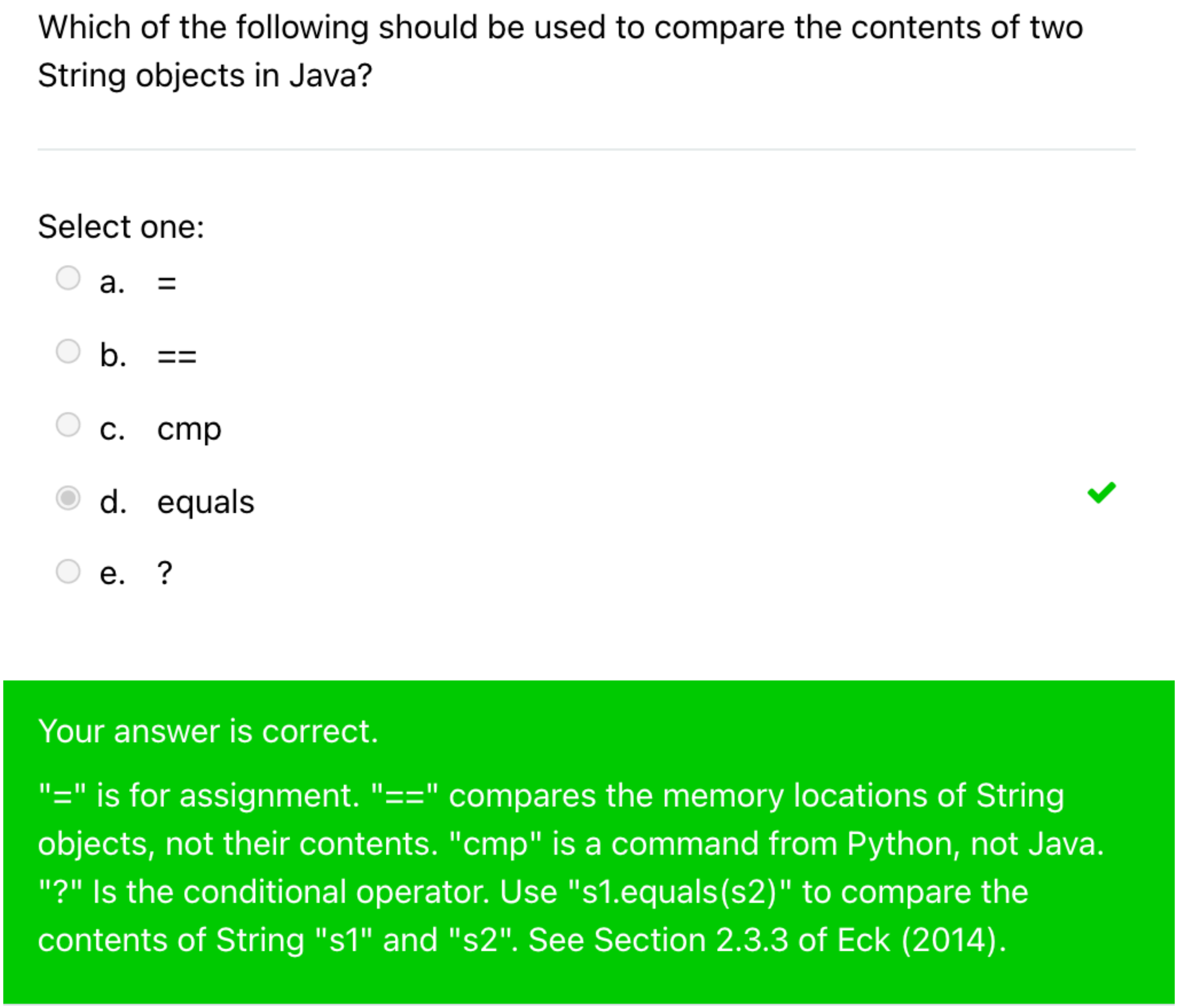
○ c. public

○ d. specification

○ e. straightforward

Your answer is correct.

Which of the following should be used to compare the contents of two String objects in Java?



Which of the following should be used to compare the contents of two String objects in Java?

Select one:

○ a. =

○ b. ==

○ c. cmp

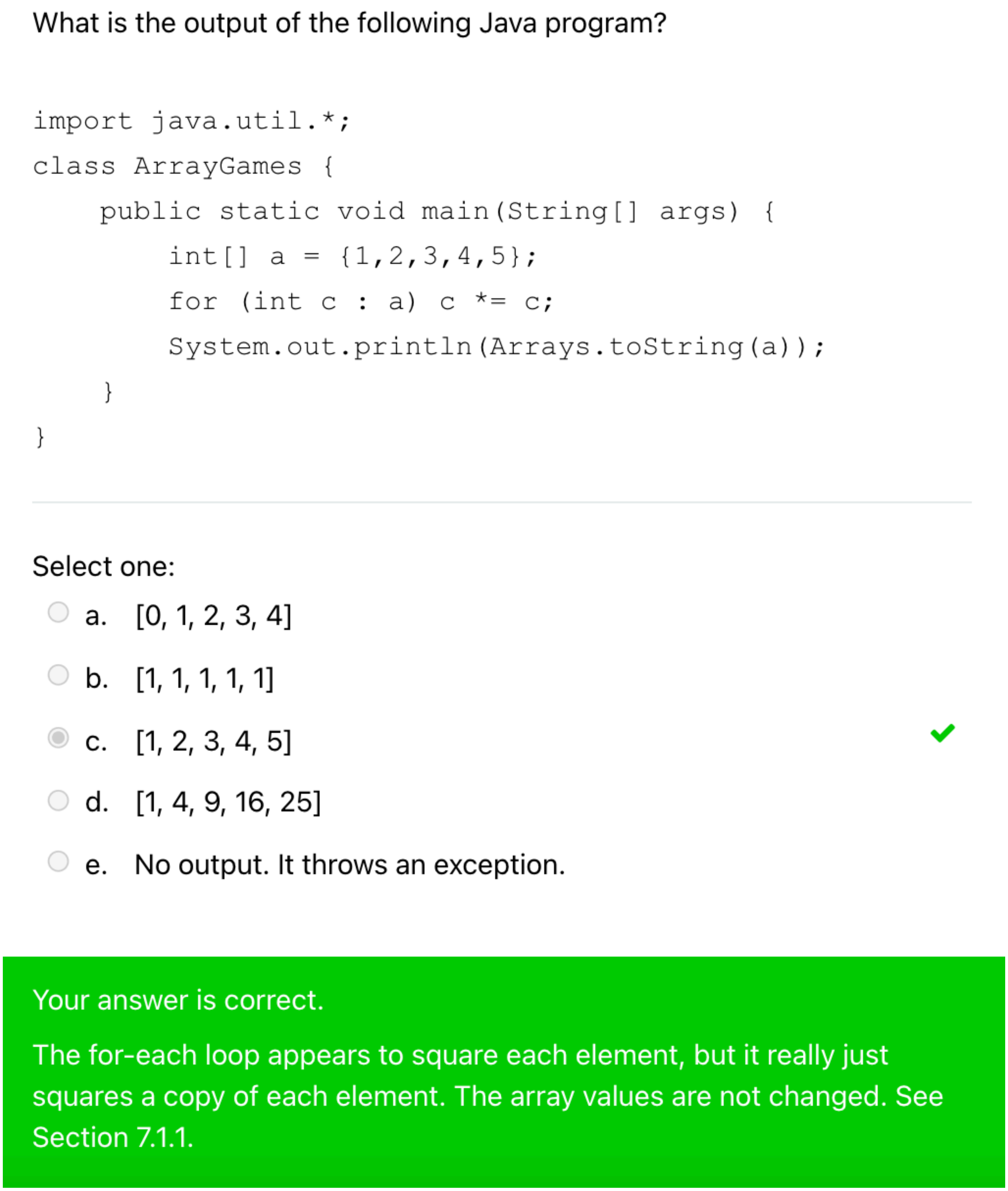
● d. equals ✓

○ e. ?

Your answer is correct.

"=" is for assignment. "==" compares the memory locations of String objects, not their contents. "cmp" is a command from Python, not Java. "?" Is the conditional operator. Use "s1.equals(s2)" to compare the contents of String "s1" and "s2". See Section 2.3.3 of Eck (2014).

What is the output of the following Java program?



What is the output of the following Java program?

```
import java.util.*;
class ArrayGames {
    public static void main(String[] args) {
        int[] a = {1,2,3,4,5};
        for (int c : a) c *= c;
        System.out.println(Arrays.toString(a));
    }
}
```

Select one:

○ a. [0, 1, 2, 3, 4]

○ b. [1, 1, 1, 1, 1]

● c. [1, 2, 3, 4, 5] ✓

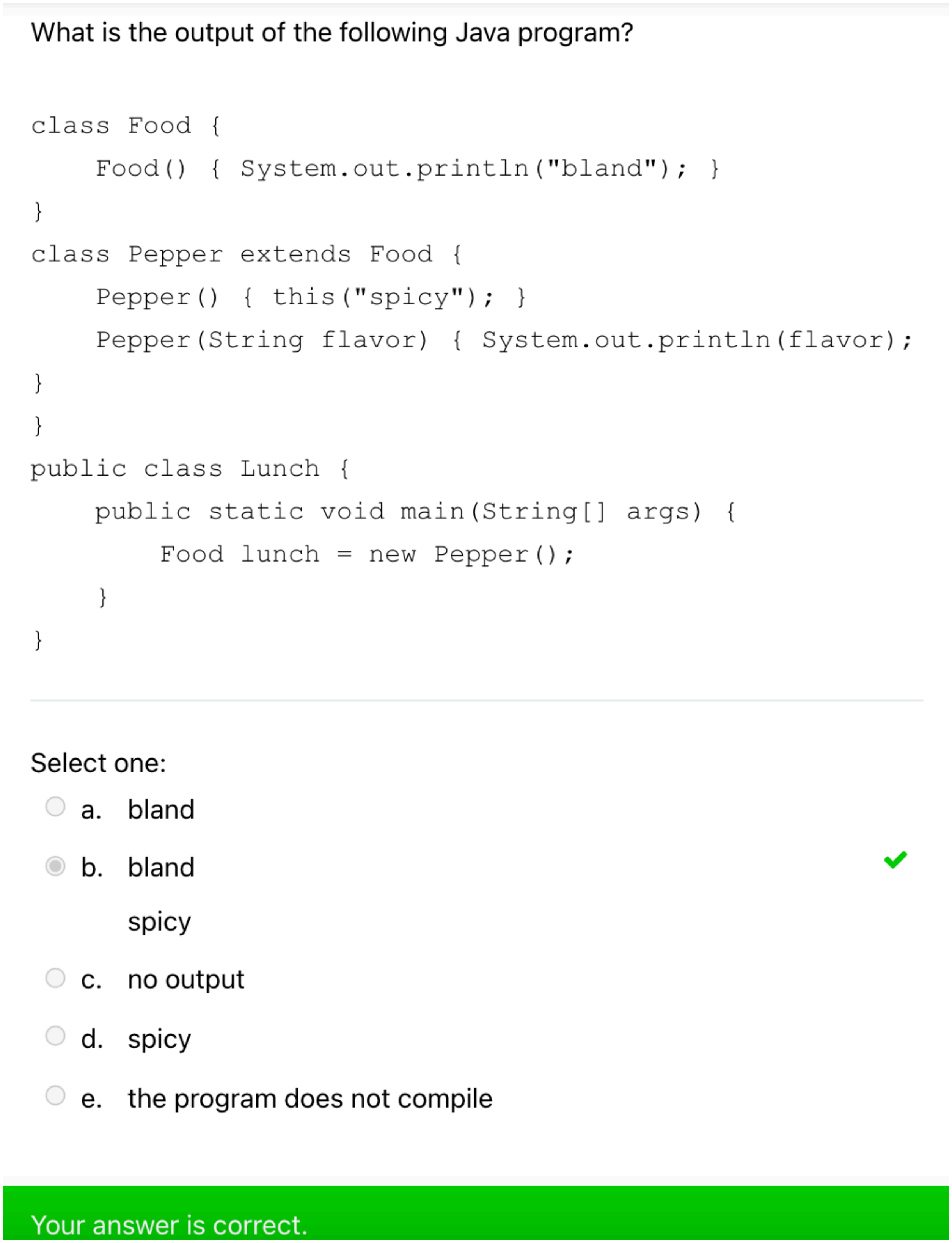
○ d. [1, 4, 9, 16, 25]

○ e. No output. It throws an exception.

Your answer is correct.

The for-each loop appears to square each element, but it really just squares a copy of each element. The array values are not changed. See Section 7.1.1.

What is the output of the following Java program?



What is the output of the following Java program?

```
class Food {
    Food() { System.out.println("bland"); }
}
class Pepper extends Food {
    Pepper() { this("spicy"); }
    Pepper(String flavor) { System.out.println(flavor);
}
}
public class Lunch {
    public static void main(String[] args) {
        Food lunch = new Pepper();
    }
}
```

Select one:

○ a. bland

● b. bland

spicy ✓

○ c. no output

○ d. spicy

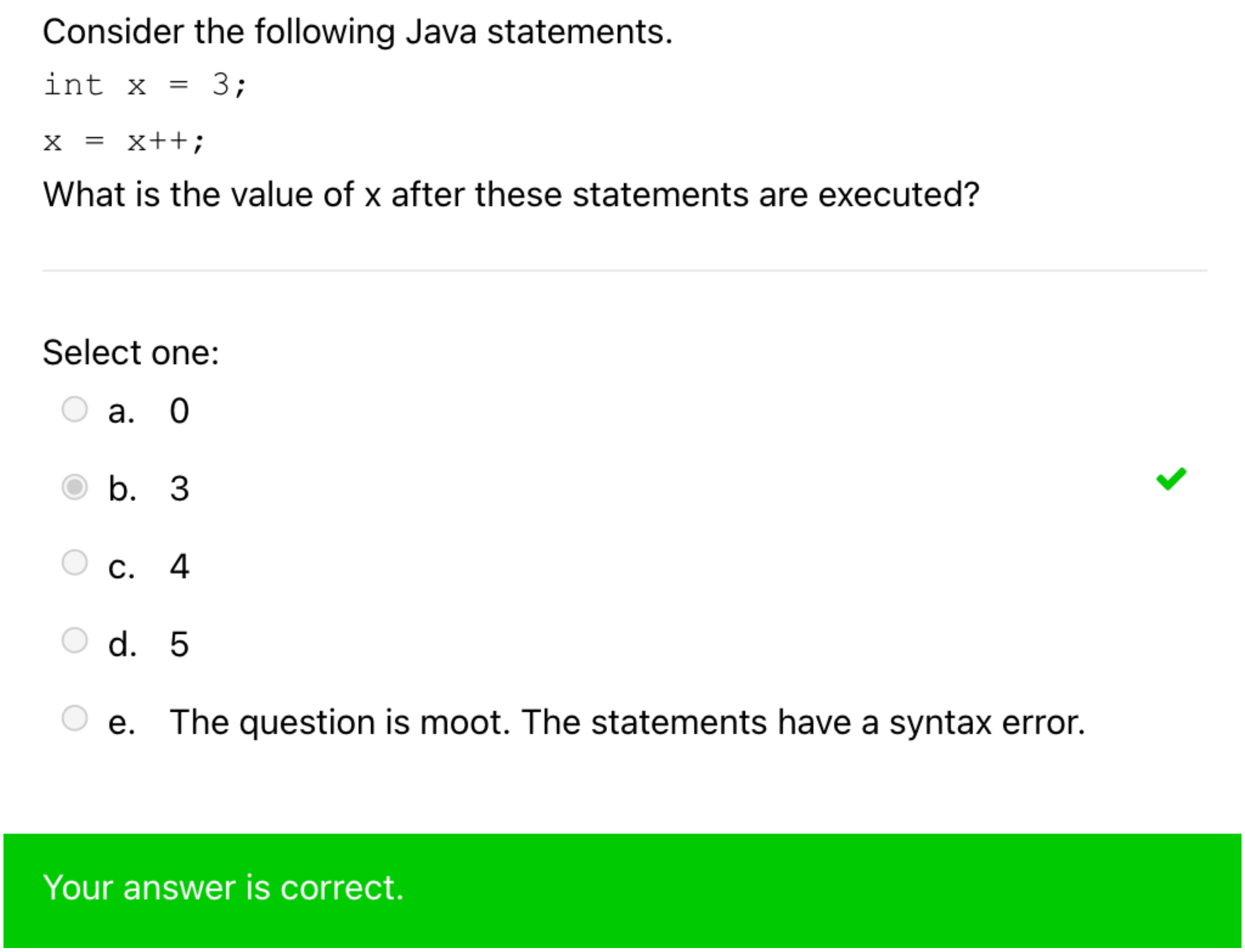
○ e. the program does not compile

Your answer is correct.

Consider the following Java statements.

```
int x = 3;
x = x++;
```

What is the value of x after these statements are executed?



Consider the following Java statements.

```
int x = 3;

x = x++;
```

What is the value of x after these statements are executed?

Select one:

○ a. 0

● b. 3 ✓

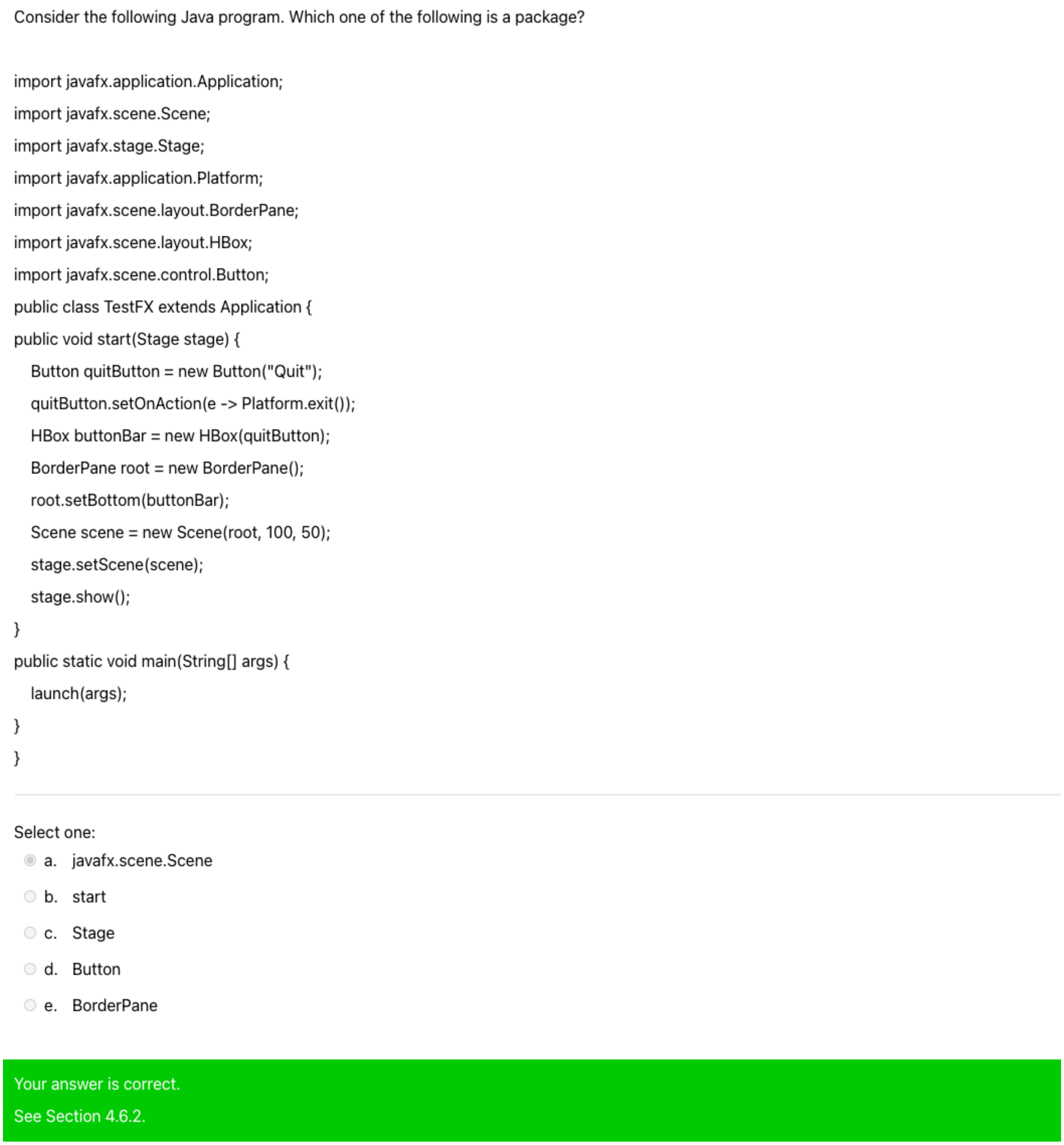
○ c. 4

○ d. 5

○ e. The question is moot. The statements have a syntax error.

Your answer is correct.

Consider the following Java program. Which one of the following is a package?



Consider the following Java program. Which one of the following is a package?

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.stage.Stage;
import javafx.application.Platform;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.control.Button;
public class TestFX extends Application {
public void start(Stage stage) {
   Button quitButton = new Button("Quit");
   quitButton.setOnAction(e -> Platform.exit());
   HBox buttonBar = new HBox(quitButton);
   BorderPane root = new BorderPane();
   root.setBottom(buttonBar);
   Scene scene = new Scene(root, 100, 50);
   stage.setScene(scene);
   stage.show();
}
public static void main(String[] args) {
   launch(args);
}
}
```

Select one:

● a. javafx.scene.Scene

○ b. start

○ c. Stage

○ d. Button

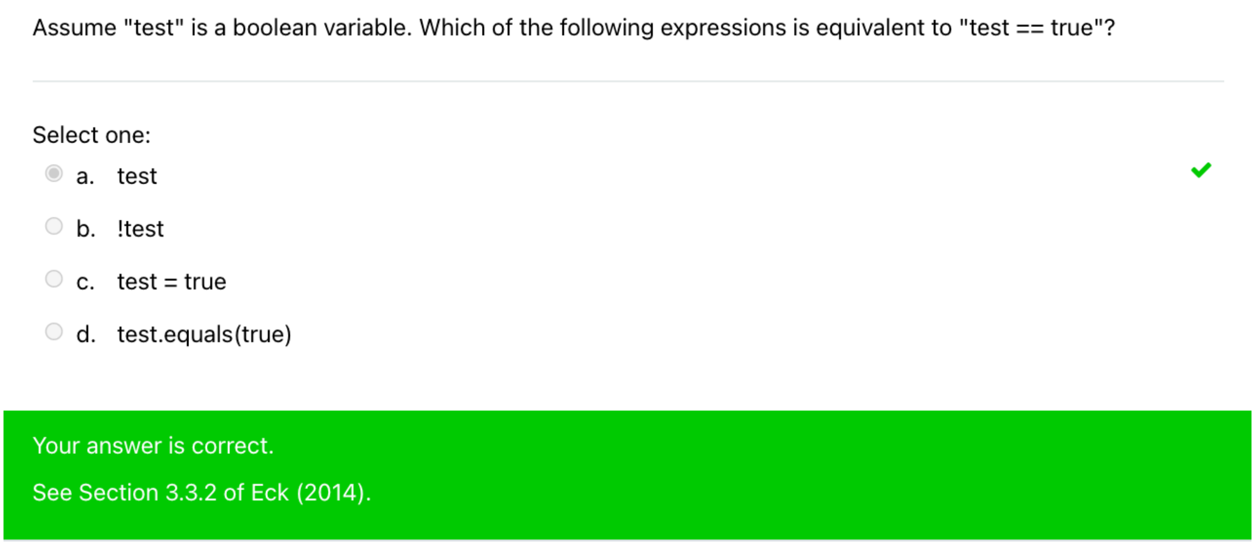
○ e. BorderPane

Your answer is correct.

See Section 4.6.2.

Answer: javafx.scene.Scene

Assume "test" is a boolean variable. Which of the following expressions is equivalent to "test == true"?



Assume "test" is a boolean variable. Which of the following expressions is equivalent to "test == true"?

Select one:

● a. test ✓

○ b. !test

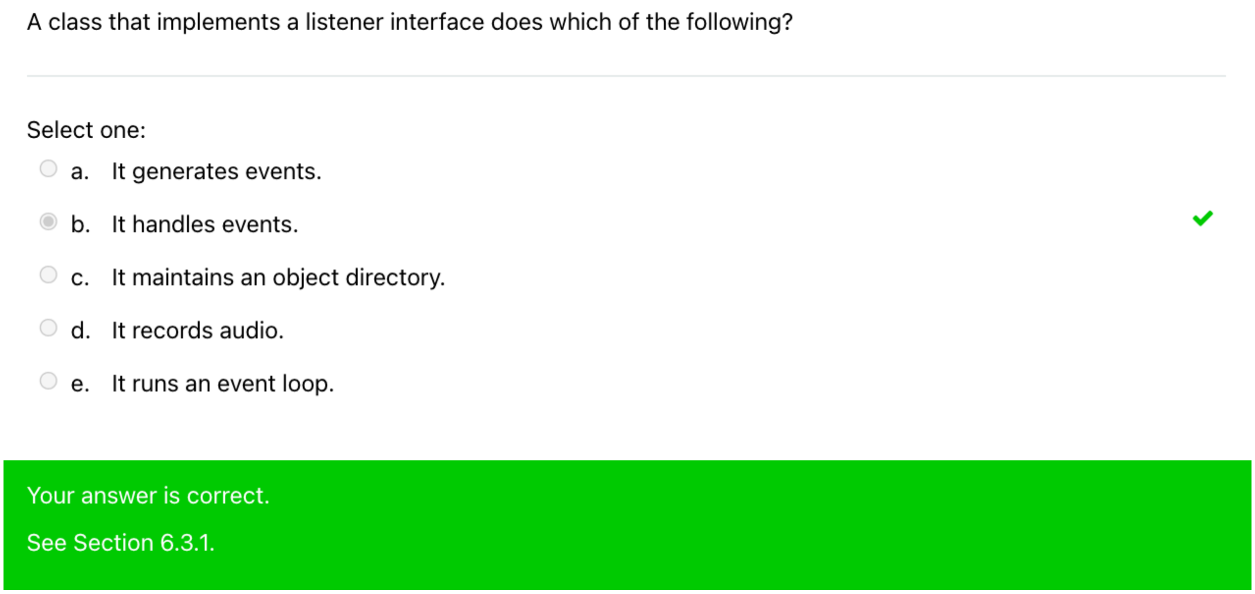
○ c. test = true

○ d. test.equals(true)

Your answer is correct.

See Section 3.3.2 of Eck (2014).

A class that implements a listener interface does which of the following?



A class that implements a listener interface does which of the following?

Select one:

○ a. It generates events.

● b. It handles events. ✓

○ c. It maintains an object directory.

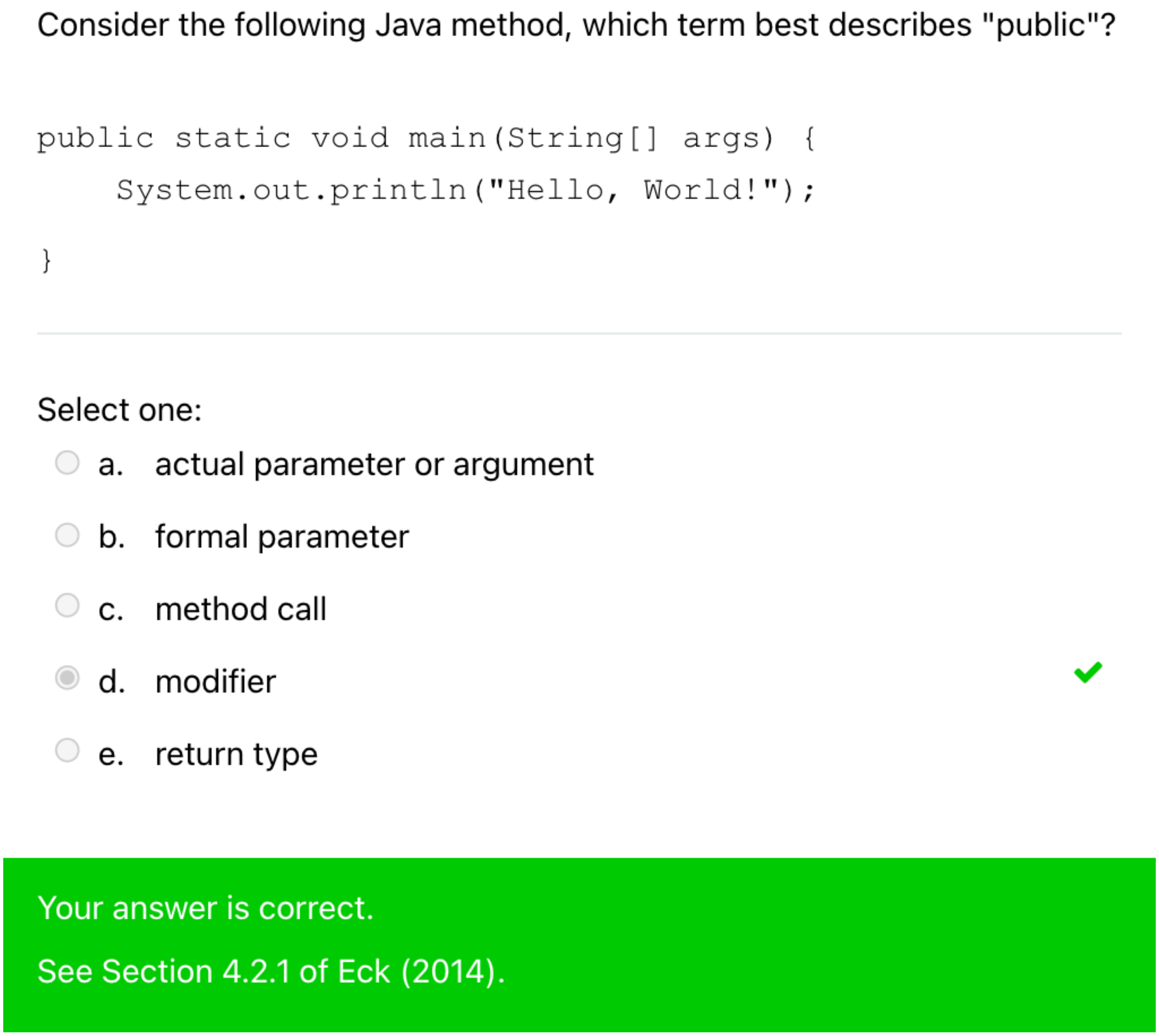
○ d. It records audio.

○ e. It runs an event loop.

Your answer is correct.

See Section 6.3.1.

Consider the following Java method, which term best describes "public"?



Consider the following Java method, which term best describes "public"?

```
public static void main(String[] args) {
    System.out.println("Hello, World!");
}
```

Select one:

○ a. actual parameter or argument

○ b. formal parameter

○ c. method call

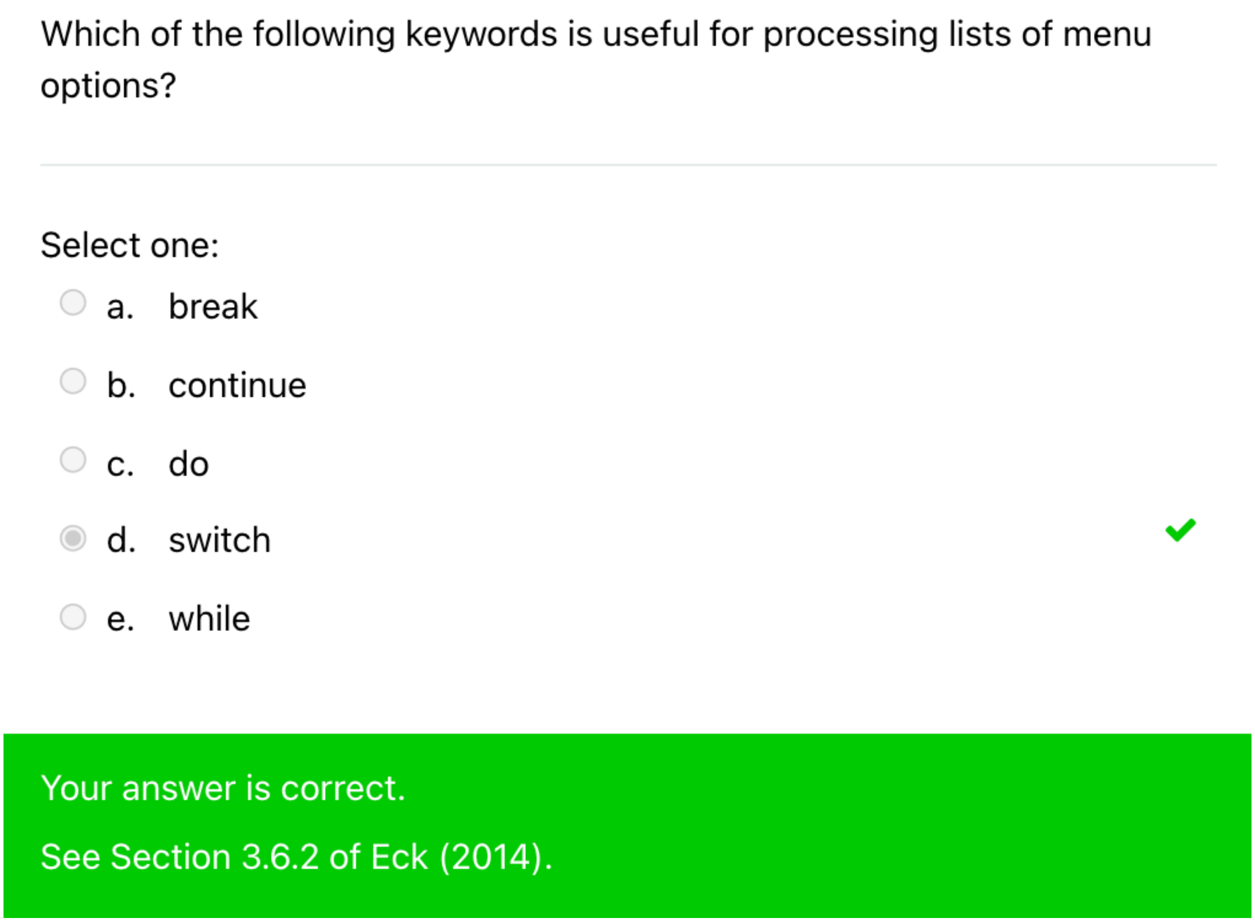
● d. modifier ✓

○ e. return type

Your answer is correct.

See Section 4.2.1 of Eck (2014).

Which of the following keywords is useful for processing lists of menu options?



Which of the following keywords is useful for processing lists of menu options?

Select one:

○ a. break

○ b. continue

○ c. do

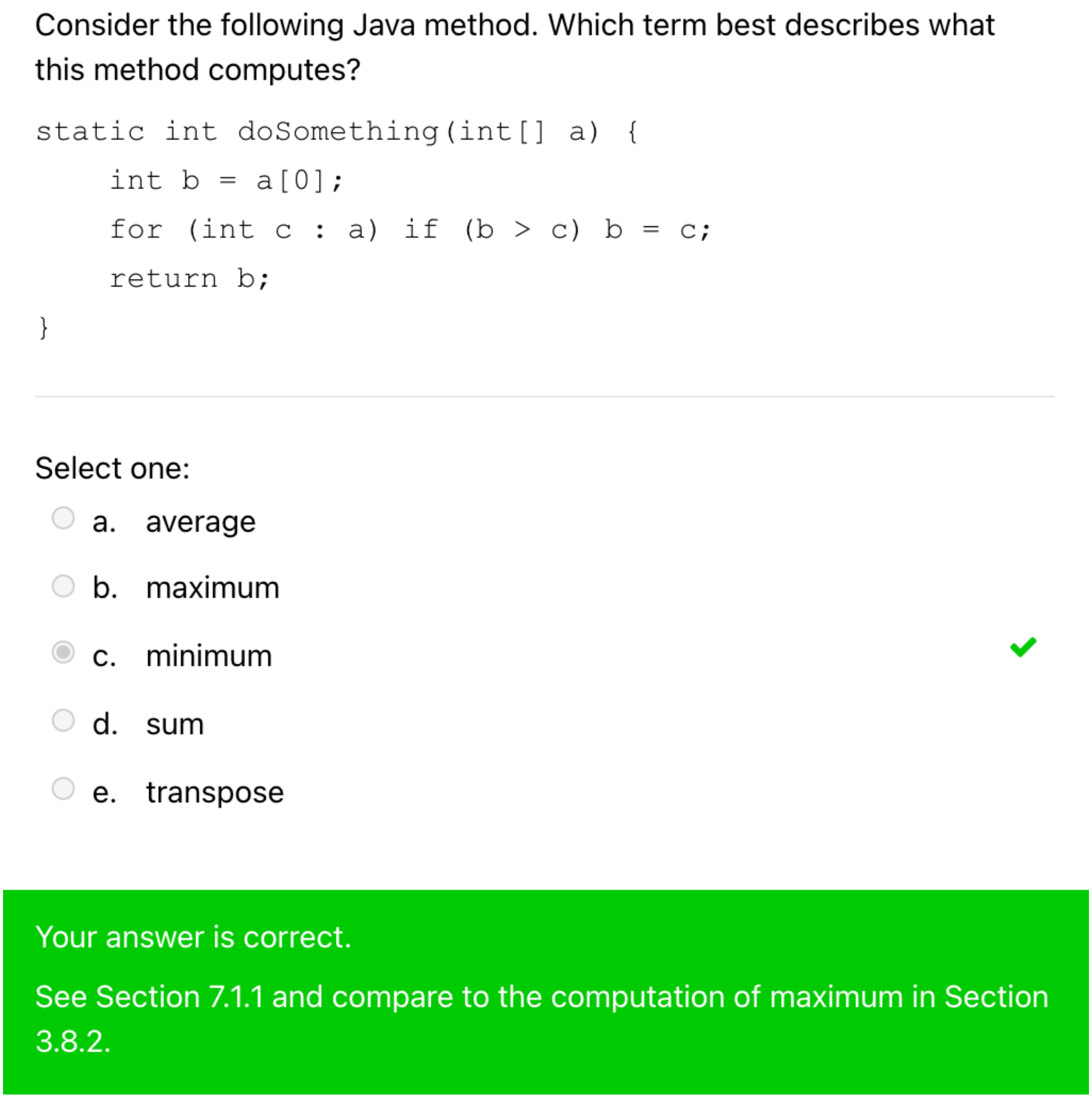
● d. switch ✓

○ e. while

Your answer is correct.

See Section 3.6.2 of Eck (2014).

Consider the following Java method. Which term best describes what this method computes?



Consider the following Java method. Which term best describes what this method computes?

```
static int doSomething(int[] a) {
    int b = a[0];
    for (int c : a) if (b > c) b = c;
    return b;
}
```

Select one:

○ a. average

○ b. maximum

● c. minimum ✓

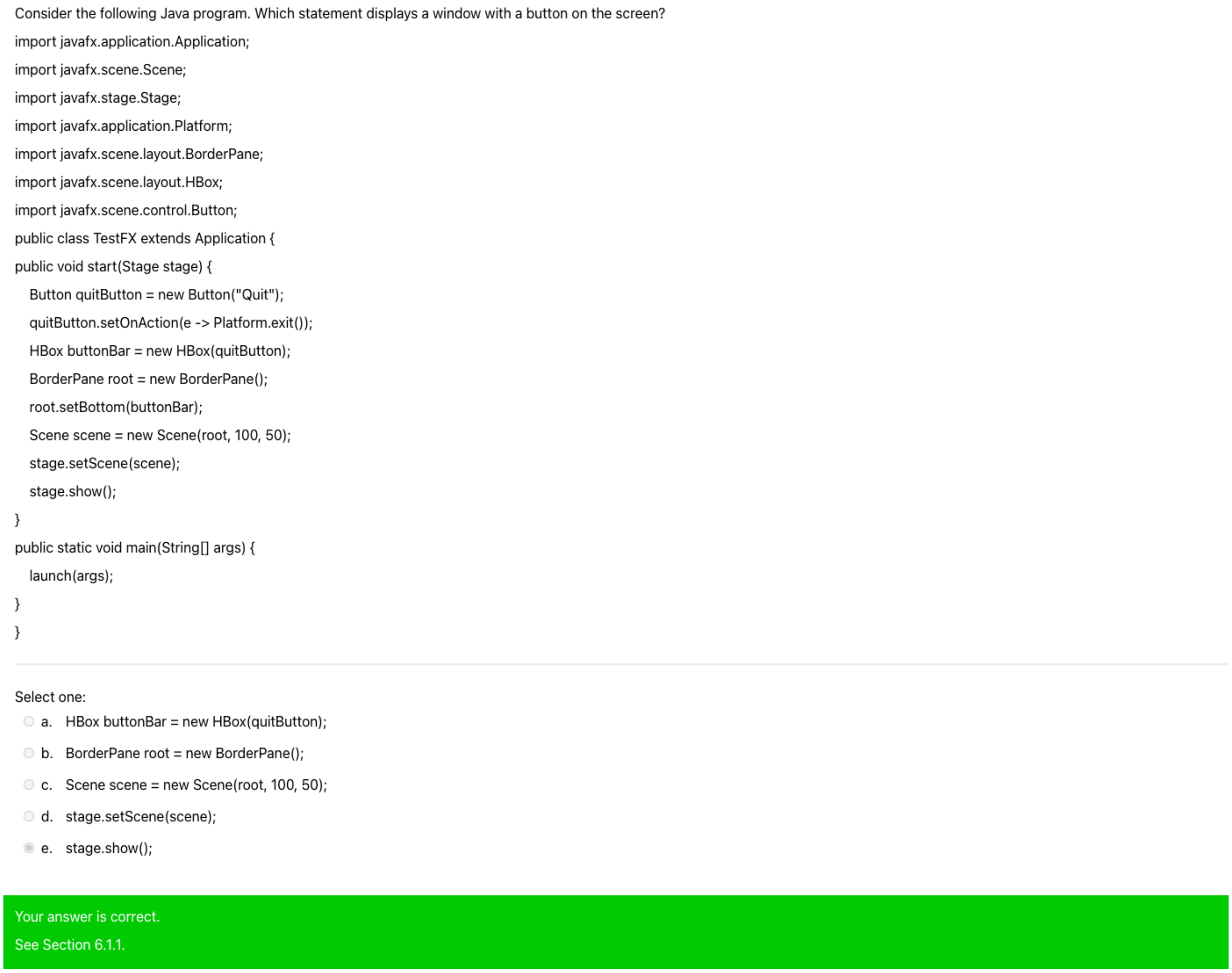
○ d. sum

○ e. transpose

Your answer is correct.

See Section 7.1.1 and compare to the computation of maximum in Section 3.8.2.

Consider the following Java program. Which statement displays a window with a button on the screen?



Consider the following Java program. Which statement displays a window with a button on the screen?

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.stage.Stage;
import javafx.application.Platform;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.control.Button;
public class TestFX extends Application {
public void start(Stage stage) {
   Button quitButton = new Button("Quit");
   quitButton.setOnAction(e -> Platform.exit());
   HBox buttonBar = new HBox(quitButton);
   BorderPane root = new BorderPane();
   root.setBottom(buttonBar);
   Scene scene = new Scene(root, 100, 50);
   stage.setScene(scene);
   stage.show();
}
public static void main(String[] args) {
   launch(args);
}
}
```

Select one:

○ a. HBox buttonBar = new HBox(quitButton);

○ b. BorderPane root = new BorderPane();

○ c. Scene scene = new Scene(root, 100, 50);

○ d. stage.setScene(scene);

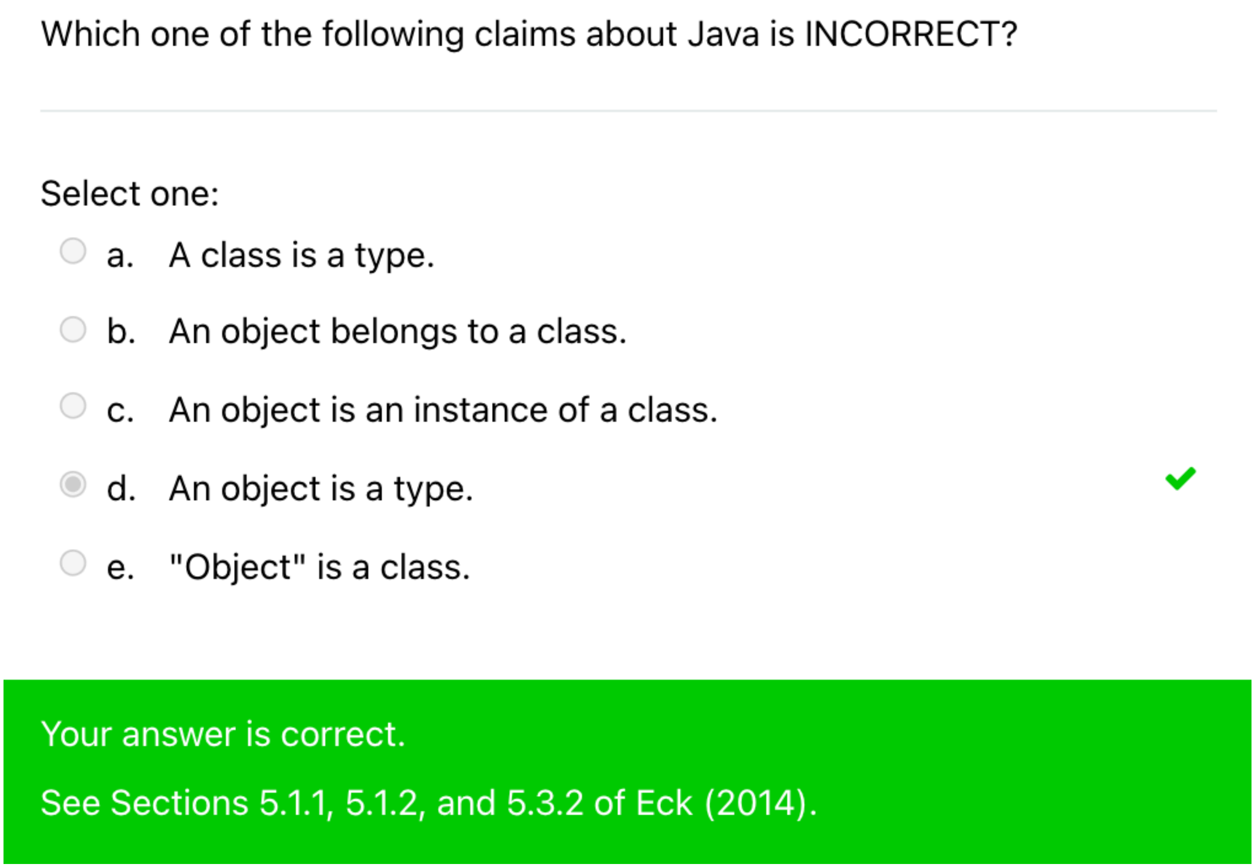
● e. stage.show();

Your answer is correct.

See Section 6.1.1.

Answer: stage.show()

Which one of the following claims about Java is INCORRECT?



Which one of the following claims about Java is INCORRECT?

Select one:

○ a. A class is a type.

○ b. An object belongs to a class.

○ c. An object is an instance of a class.

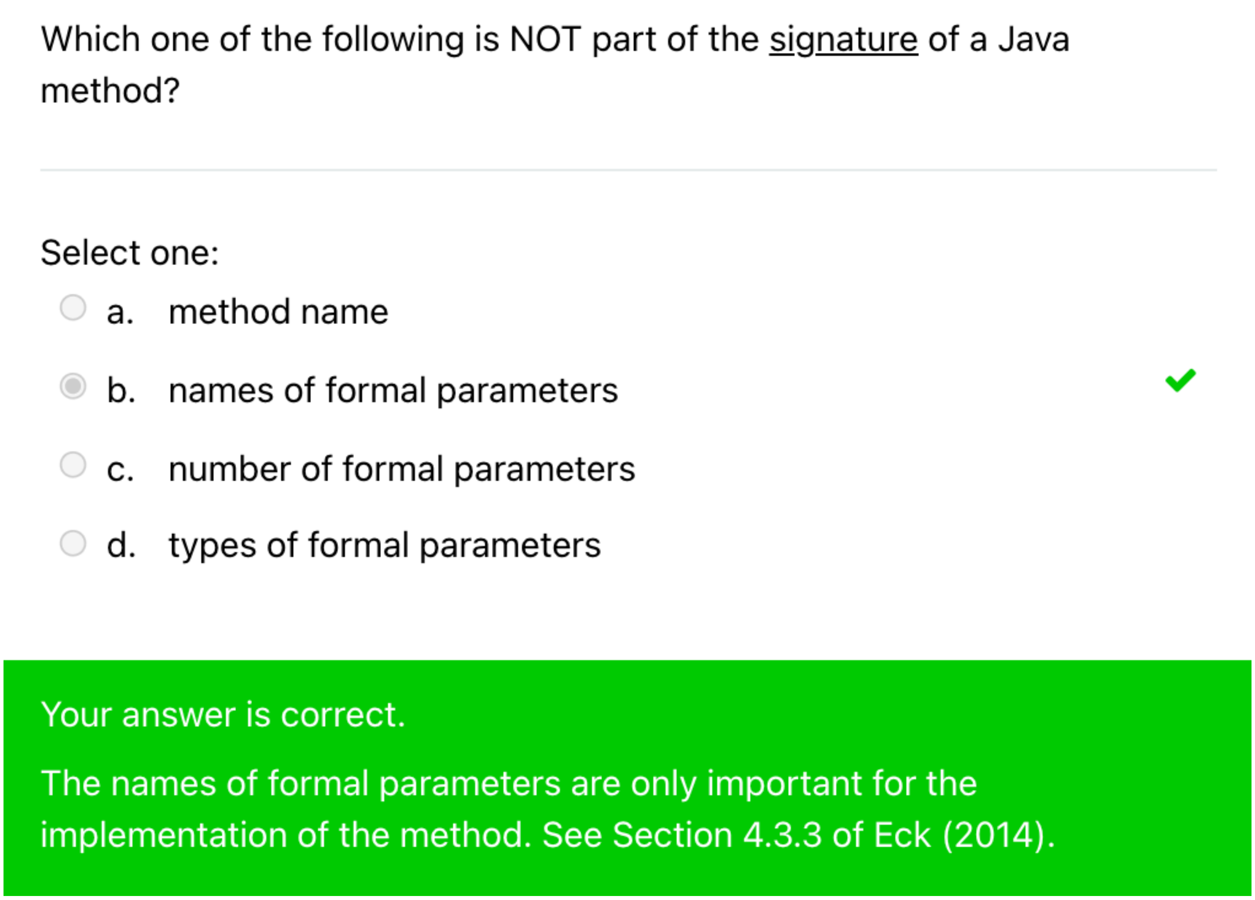
● d. An object is a type. ✓

○ e. "Object" is a class.

Your answer is correct.

See Sections 5.1.1, 5.1.2, and 5.3.2 of Eck (2014).

Which one of the following is NOT part of the signature of a Java method?



Which one of the following is NOT part of the signature of a Java method?

Select one:

○ a. method name

● b. names of formal parameters ✓

○ c. number of formal parameters

○ d. types of formal parameters

Your answer is correct.

The names of formal parameters are only important for the implementation of the method. See Section 4.3.3 of Eck (2014).

Consider the following Java method. Which term best describes what this method computes?

Consider the following Java method. Which term best describes what this method computes?

```
static void doSomething(int[][] a) {
    int n = a.length;
    for (int j = 0; j < n; j++) {
        for (int i = j+1; i < n; i++) {
            int aij = a[i][j];
            a[i][j] = a[j][i];
            a[j][i] = aij;
        }
    }
}
```

Select one:

○ a. average

○ b. maximum

○ c. minimum

○ d. sum

● e. transpose ✓

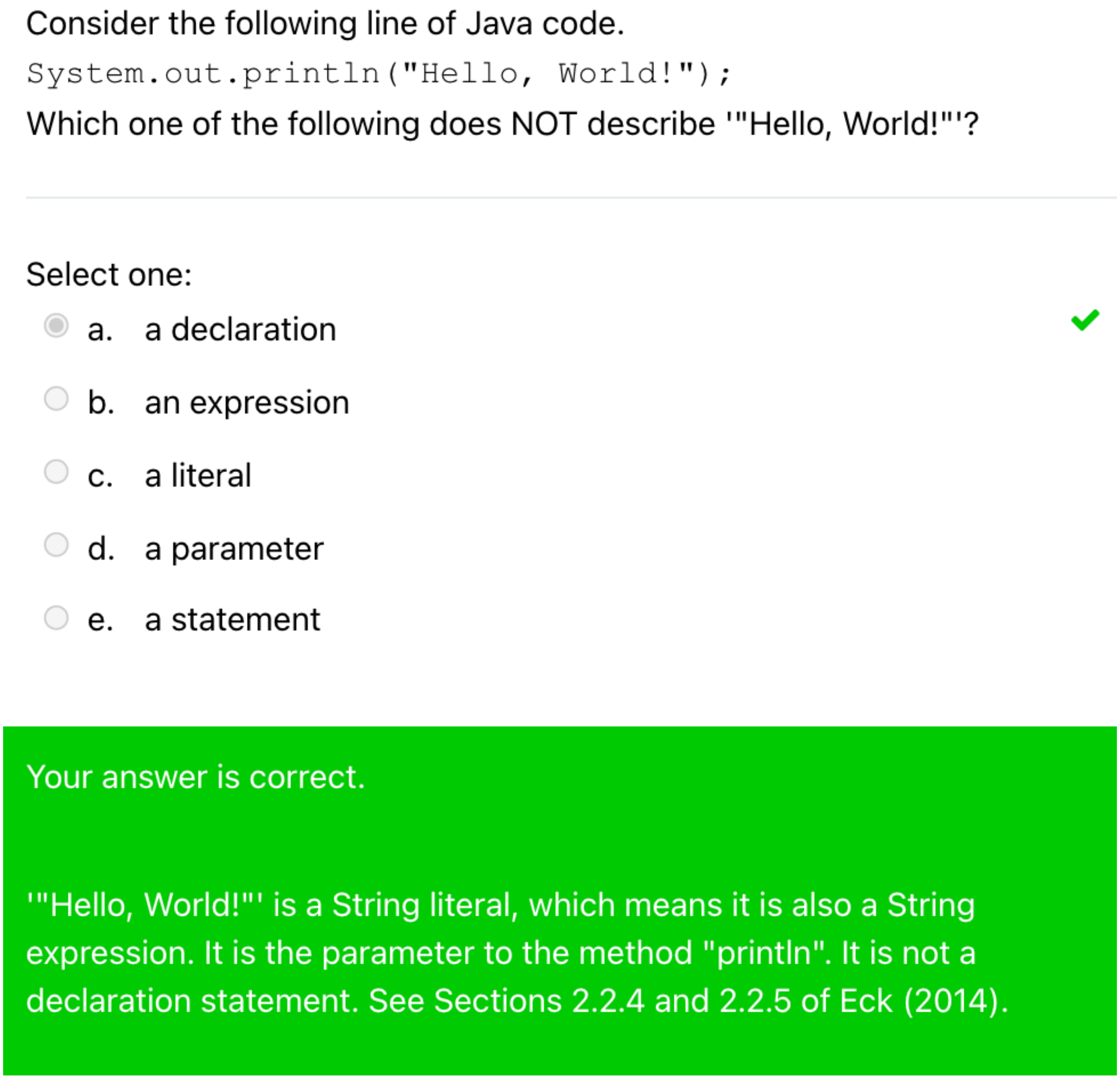
Your answer is correct.

See Section 7.5.1 and Chapter 7 Exercise 2.

Consider the following line of Java code.

```
System.out.println("Hello, World!");
```

Which one of the following does NOT describe '"Hello, World!"'?



Consider the following line of Java code.

```
System.out.println("Hello, World!");
```

Which one of the following does NOT describe '"Hello, World!"'?

Select one:

● a. a declaration ✓

○ b. an expression

○ c. a literal

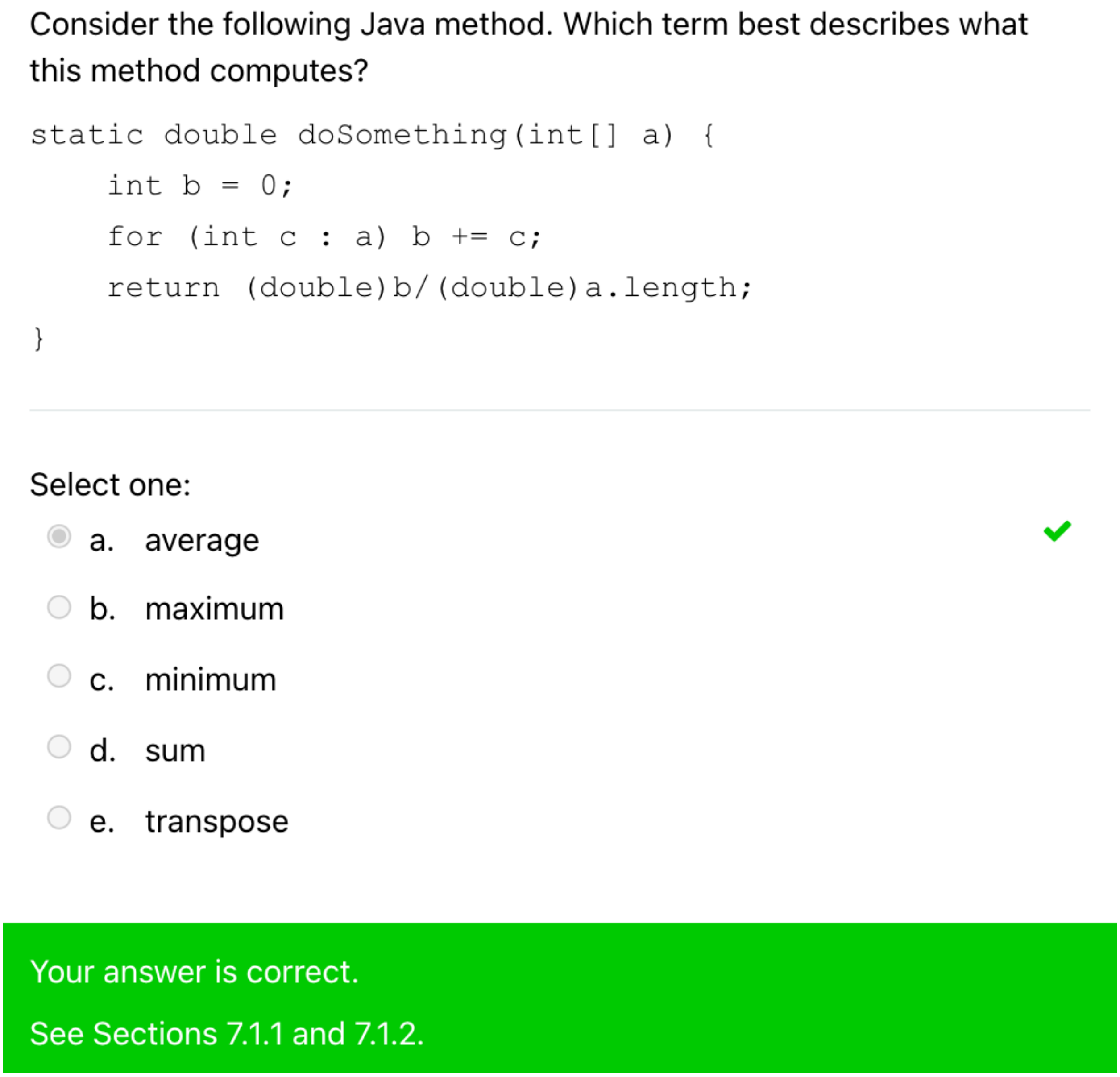
○ d. a parameter

○ e. a statement

Your answer is correct.

'"Hello, World!"' is a String literal, which means it is also a String expression. It is the parameter to the method "println". It is not a declaration statement. See Sections 2.2.4 and 2.2.5 of Eck (2014).

Consider the following Java method. Which term best describes what this method computes?



Consider the following Java method. Which term best describes what this method computes?

```
static double doSomething(int[] a) {
     int b = 0;
     for (int c : a) b += c;
     return (double)b/(double)a.length;
}
```

Select one:

● a. average ✓

○ b. maximum

○ c. minimum

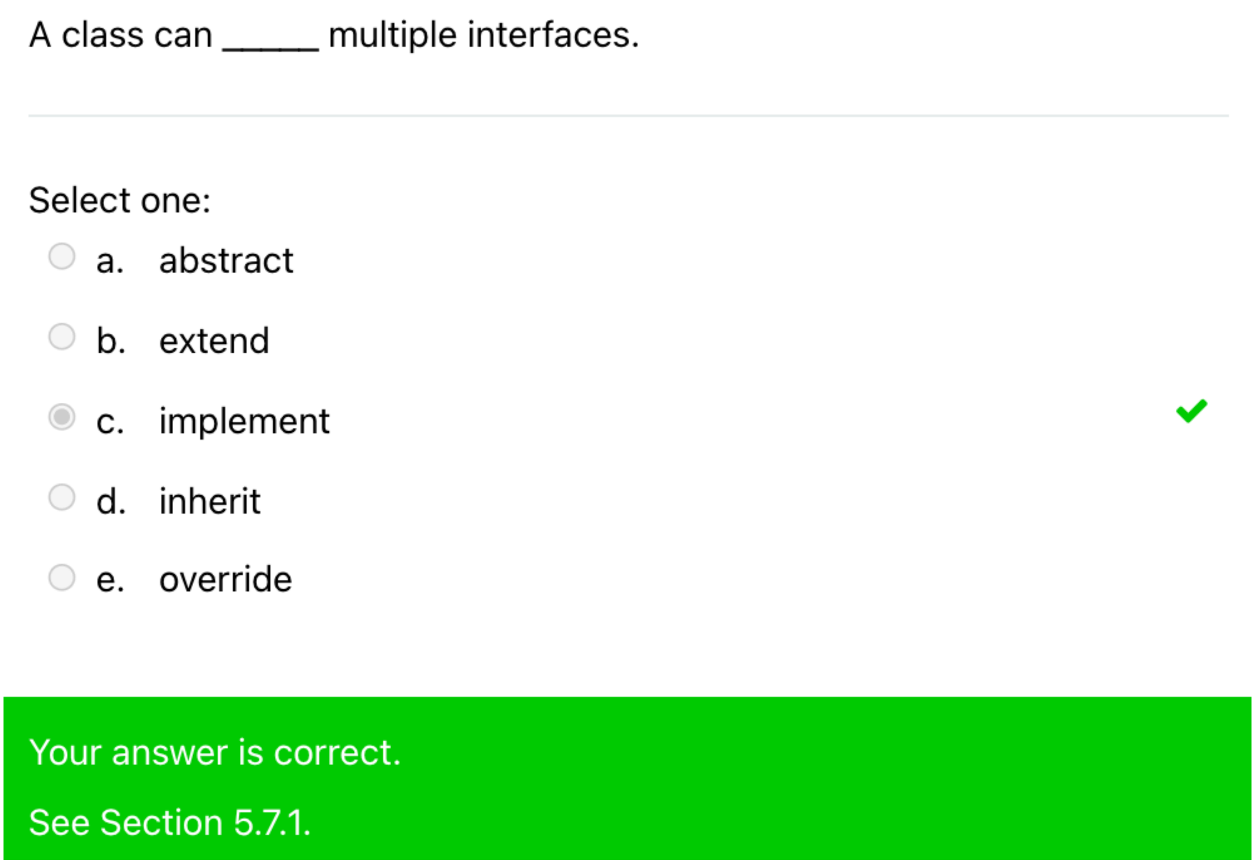
○ d. sum

○ e. transpose

Your answer is correct.

See Sections 7.1.1 and 7.1.2.

A class can _____ multiple interfaces.



A class can ______ multiple interfaces.

Select one:

○ a. abstract

○ b. extend

● c. implement ✓

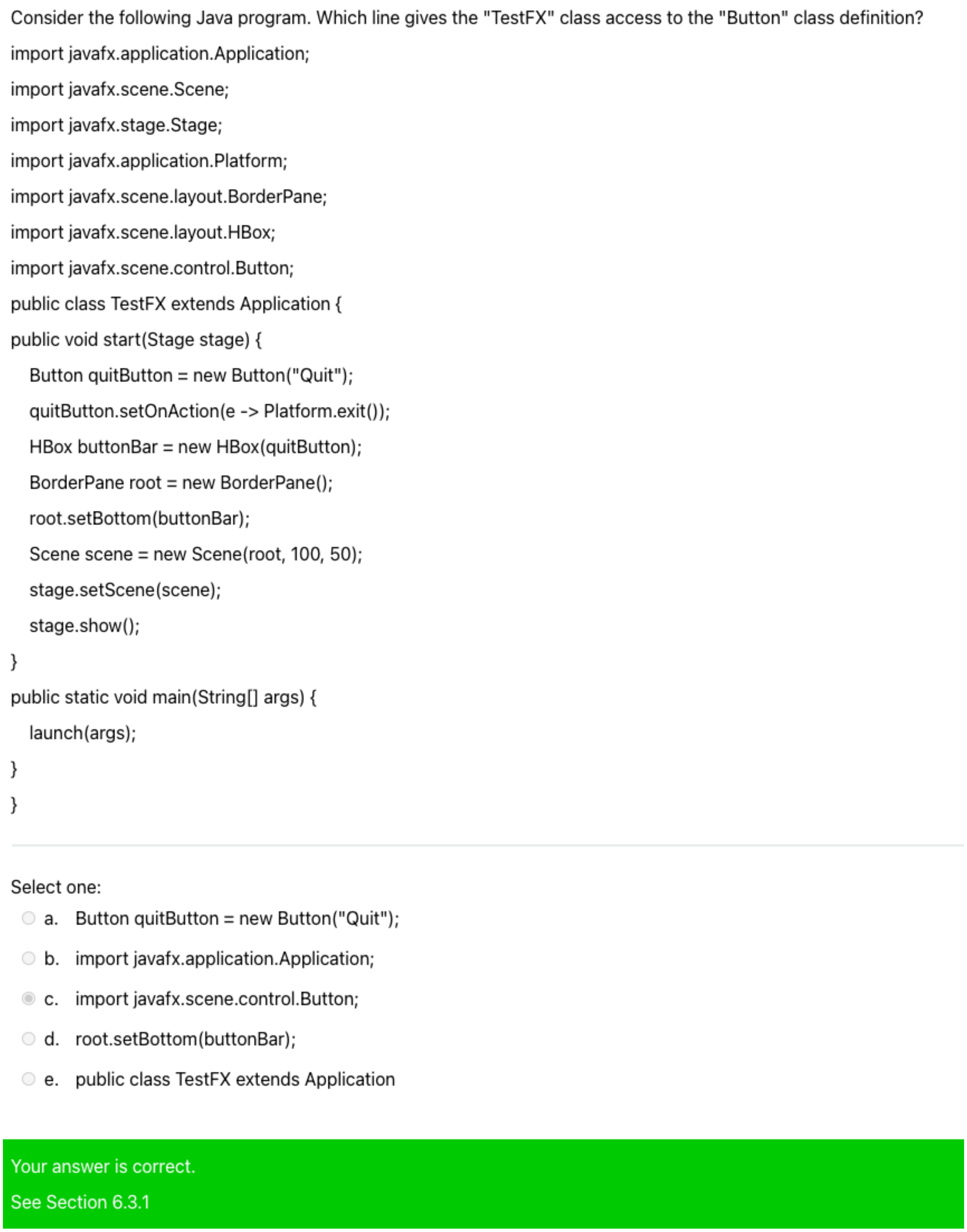
○ d. inherit

○ e. override

Your answer is correct.

See Section 5.7.1.

Consider the following Java program. Which line gives the "TestFX" class access to the "Button" class definition?



Consider the following Java program. Which line gives the "TestFX" class access to the "Button" class definition?

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.stage.Stage;
import javafx.application.Platform;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.control.Button;
public class TestFX extends Application {
public void start(Stage stage) {
   Button quitButton = new Button("Quit");
   quitButton.setOnAction(e -> Platform.exit());
   HBox buttonBar = new HBox(quitButton);
   BorderPane root = new BorderPane();
   root.setBottom(buttonBar);
   Scene scene = new Scene(root, 100, 50);
   stage.setScene(scene);
   stage.show();
}
public static void main(String[] args) {
   launch(args);
}
}
```

Select one:

○ a. Button quitButton = new Button("Quit");

○ b. import javafx.application.Application;

● c. import javafx.scene.control.Button;

○ d. root.setBottom(buttonBar);

○ e. public class TestFX extends Application

Your answer is correct.

See Section 6.3.1

Answer: import javafx.scene.control.Button;