ANSWER KEY

Answer Key: Day 3

Every page for this day is keyed, and nothing is left as "answers will vary." This file covers the bell ringer, the guided notes and the exit ticket. In this folder, the key for the 1.1 Quiz Task sheet is in Activity_and_Lab_Keys.pdf and the lesson quiz has its own key.

Bell Ringer: Get Set, Go

```
1  System.out.print("Get ");
2  System.out.println("set.");
3  System.out.print("Go!");
```

Look for

- Question 1: 2 lines.
- Question 2: no. Listen for students who say `println` on line 2 ends the line.

Model response

It prints 2 lines, so it doesn't do what Ahmed wants. `println` on line 2 ends the line.

Guided Notes: Day 3

Blanks are shown filled in, in order.

Review: print and println

`System.out.print()` prints its text and stays on the same line.

`System.out.println()` prints its text and then starts a new line.

To print two pieces of text on one line, use `print` for the first piece.

Review: compile errors and the parts of a program

```java
// A note for the class
public class Main {
    public static void main(String[] args) {
        System.out.println("Quiz day!");
    }
}
```

Output

```
Quiz day!
```

Java cares about capital letters. `system.out` with a small s is a **compile error**.

Java can't read past a missing `;`, so it may show only that first error.

Fix it, then **compile** again.

A **comment** is a note for people. Java skips it.

The **class** holds the **main method**, and Java starts running there.

Exit Ticket

The exit ticket is worth 3 points: 1 for each item.

1. Correct: (A) 1 word: blue

`args` *holds only the words typed after the program's name, so it holds blue.*

2. Correct: (A) Line 1, the comment

```java
// Print the menu.
System.out.print("Menu");
```

Line 1 starts with `//`, so it's a comment. Java skips comments.

Item 3: written answer

Look for

- Java showed the missing semicolon first. It stops at a mistake it can't read, so the `printLn` error showed only after that fix.

Model response

Java showed the missing semicolon first. Java can't read past that mistake, so it showed the `printLn` error only after I fixed the semicolon and compiled again.