ANSWER KEY

Answer Key: Day 1

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 What's in args? sheet is in Activity_and_Lab_Keys.pdf.

Bell Ringer: Beatrice's First Lines

```
1  System.out.print("Hello, ");
2  System.out.println("Beatrice!");
3  System.out.print("Welcome to class.");
```

Look for

- Question 1: any number. Listen for students who say 3 because the code has three lines.
- Question 2: any reasoned guess. The text of lines 1 and 2 prints on one line, because `print` doesn't end the line.

Model response

It prints 2 lines. Line 1 uses `print`, so the text of line 2 goes on the same line. Line 2 uses `println`, so the text of line 3 starts a new line.

Guided Notes: Day 1

Blanks are shown filled in, in order.

The parts of a Java program

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

Output

```
Hello!
```

Java code lives in a **class**. `public class Main {` starts one named `Main`, saved in a file named `Main.java`.

Java starts running the program in the **main method**. Its first line is `public static void main(String[] args) {`.

Braces mark what is inside. Each `{` opens a part, and its `}` closes it.

Words from the command line

The **command line** is where you type a command to run a program, like java Main.

The words typed after java Main are called arguments. `String[] args` holds them.

The command java Main gives `args` no words. The command java Main red 7 gives `args` 2 words: red and 7.

Statements, semicolons and comments

```java
// Ahmed's to-do list
System.out.println("Feed the cat.");
/* System.out.println("Nap."); */
System.out.println("Read.");
```

Output

```
Feed the cat.
Read.
```

A **statement** is one step in a program, like one print line.

A print line ends with a **semicolon**, the `;` mark. Java doesn't end a statement at a line break.

A **comment** is a note for people, and Java skips it. `//` starts one that runs to the end of the line.

A block comment starts with `/*` and ends with `*/`. It can turn off a line of code, like line 3 above.

Printing text

```
1  System.out.print("Hello, ");
2  System.out.println("Beatrice!");
3  System.out.print("Welcome to class.");
```

Output

```
Hello, Beatrice!
Welcome to class.
```

Text to print goes inside double quotes. The quotes don't print.

`System.out.println()` prints its text and then ends the line.

`System.out.print()` prints its text and stays on the same line. The next text prints right after it.

Write what this code prints in the box.

Exit Ticket

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

1. Correct: (B) At the `main` method

Java starts running every program in the `main` method.

2. Correct: (C) 2

```
1  System.out.print("Ready ");
2  System.out.println("now.");
3  System.out.println("Go!");
```

Line 1 uses `print`, so now. prints on the same line. Line 2 ends that line, and Go! prints on a second line.

Item 3: written answer

Model response

No. `println` prints its text and ends the line, but `print` stays on the same line, so the next text prints right after it.

Look for this idea in the student's own words.