

Topic 1.1: Introduction to Algorithms, Programming, and Compilers

LO: 1.1.A, 1.1.B, 1.1.C | Skill: 1.A | Scenario: Algorithms in everyday life — then your first Java program

Scenario — Riverside Coding Club

Priya just joined the after-school coding club at Riverside High. Her first task is to write a short Java program that welcomes members and figures out how many water bottles the club needs for a weekend hike. Read what happens, then answer the questions.

Step 1 — Priya types her first version

She writes this and clicks Run. The IDE refuses to run it and underlines a line in red.

Version 1 — ClubWelcome.java

```
public class ClubWelcome {
    public static void main(String[] args) {
        System.out.println("Welcome to the Riverside Coding Club!")
    }
}
```

Step 2 — She fixes that and the program runs

Priya adds the missing semicolon. Now it runs, and she adds the bottle calculation below. It prints 16, but the club only has 12 hikers — that cannot be right.

Version 2 — bottle calculation

```
int hikers = 12;
int perBottle = 4;           // each bottle serves 4 hikers
int bottlesNeeded = hikers + perBottle;
System.out.println(bottlesNeeded); // prints 16, but the club has 12 hikers
```

Trace the three variables to see where 16 comes from:

Line executed	hikers	perBottle	bottlesNeeded	Output
int hikers = 12;	12	—	—	—
int perBottle = 4;	12	4	—	—
int bottlesNeeded = hikers + perBottle;	12	4	16	—
System.out.println(bottlesNeeded);	12	4	16	16

Step 3 — A later version crashes

Priya rewrites the calculation to split hikers into groups. During a test where no groups have been entered yet, the program stops with an error message and quits.

Version 3 — splitting into groups

```
int hikers = 12;
int groups = 0;           // no groups were entered yet
int perGroup = hikers / groups; // divide by zero at run time
System.out.println(perGroup);
```

Questions

1. In Version 1, what type of error stopped the program from running, and which tool reported it?

Model: A syntax error (the missing semicolon). The compiler reported it, which is why the program would not run until she fixed it.

2. In Version 2, the program runs but prints the wrong number. What type of error is this, and how did Priya discover it?

Model: A logic error. The code adds hikers and perBottle instead of dividing, so it runs but gives a wrong result. She discovered it by testing — comparing the output (16) to a value she knew was reasonable.

3. In Version 3, the program quits during execution. What type of error is this, and why did the compiler not catch it?

Model: A run-time error (an exception): dividing by zero. The compiler cannot know that groups will be 0 when the program runs, so it only fails during execution.

4. Priya says, "If it compiles, it must be correct." Use this scenario to explain why she is wrong.

Model: Compiling only means the syntax is valid. Version 2 compiled and ran but still gave the wrong answer (logic error), and Version 3 compiled but crashed at run time. Passing the compiler is necessary but not sufficient — you still have to test.