

COLLABORATION ACTIVITY

Topic 1.1: Introduction to Algorithms, Programming, and Compilers

Collaboration Activity — Algorithms & Bugs

Work in teams of 3-4. You have two missions: design an algorithm, and play 'bug detective' with three Java segments. Assign the roles below, then complete both missions on chart paper or a shared doc.

Team roles

Role	Responsibility
Facilitator	Keeps the team on time and makes sure every voice is heard.
Recorder	Writes the team's algorithm and final bug verdicts.
Tester	Walks through each algorithm/segment with a sample input to check it.
Reporter	Shares the team's reasoning with the class during debrief.

Mission 1 — Design an algorithm

Choose ONE everyday task (lining up for lunch, charging phones for a field trip, or sorting books onto a cart). As a team, write a numbered-step algorithm precise enough that another team could follow it with no guessing. Then have your Tester run it with a sample and mark any step where sequencing matters.

Mission 2 — Bug detective

For each segment, agree on the error type (syntax, logic, or run-time) and write one sentence on how it would be discovered. Defend your verdict to the class.

Exhibit A

Exhibit A

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

Exhibit B

Exhibit B

```
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
```

Exhibit C

Exhibit C

```
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);
```

Team deliverable — First-Program Checklist

Combine what you learned into a 4-item checklist a brand-new programmer should follow between writing code and trusting its output.

DISCUSSION CLOSE

Each Reporter shares one checklist item and the one exhibit the team found hardest to classify. As a class, decide which error type is the most dangerous to miss, and why.