Broken Space Robot Debugging Mission: Rebooting R-B0T of Planet *JAVA*

Welcome, Junior Engineers from Earth!

R-B0T, our friendly exploration robot of planet *JAVA*, has suffered a catastrophic software corruption. Its brain (the Java source code) is full of bugs, illegal instructions, and broken logic. Your mission is to **debug, repair, and reboot** the robot so it can walk, sense, calculate, and collect items again.

Compile often. Let the compiler guide you. Every error is a clue.

The Story: System Failure on Planet JAVA

During a solar storm, R-B0T's software was corrupted. Now it can't even start properly.

Your job is to repair its brain by fixing the broken code inside **`App.java`** so the robot can:

1. Boot correctly
2. Identify itself
3. Read sensors
4. Manage power safely
5. Store collected items
6. Calculate distances
7. Communicate status messages
8. Walk forward without getting stuck forever

The Broken Brain

Open **`src/main/java/main/App.java`** and replace its contents with the attached corrupted code. Everything happens inside `App.java`.

Your Mission Tasks

Fix **all system failures** so the project compiles and runs correctly.

1. Repair the constructor so the robot boots correctly
2. Fix illegal variable names and reserved words
3. Repair broken casting and data types
4. Replace the broken inventory with a dynamic structure
5. Fix the distance calculator method and its return value
6. Repair the status message method
7. Fix the circle area calculator using constants and operators
8. Fix the memory test loop
9. Stop the infinite walking loop
10. Fix the main method so the program runs correctly

Expected Final Behavior (After Fixing)

When everything is repaired, running the program should:

1. Print a boot message
2. Display a clean status message
3. Calculate and display a distance
4. Store at least three items and print the inventory size
5. Calculate a circle area correctly
6. Call **memoryTest()** function exactly once
7. Walk forward exactly five steps and stop