# CommandBot

You are designing a small robot that moves on a 2D grid. The robot starts at position (0,0) facing **NORTH**. It can **MOVE** forward or backward, **TURN** by 90-degree increments, and **REPORT** its position. Each command will be represented by a class that implements a **Command** interface. The robot will extend a base class that provides basic move and turn functionality. You will complete **three classes:** CommandBot, Command, and RobotPart.

**Write the Command interface. Include:**

- Three methods named `apply`, `inverse`, and `name`.
- Each command will implement this interface to perform its own action.

**Write the RobotPart abstract class. Include:**

- One private field for the part's name
- A constructor that sets the name
- A method `requireHealthy()` that throws an exception if the part is not working
- An abstract method `healthy()` that returns whether the part is functioning

**Write the CommandBot class that extends RobotBase. Include:**

- An **array** to store executed commands
- Methods `execute`, `executeAll`, and `returnToOrigin`
- Inner classes `Move`, `Turn`, and `Report` that each implement **Command**
- The robot must return to **(0,0,NORTH)** when `returnToOrigin()` is called by reversing the stored commands

Write the complete CommandBot, Command, and RobotPart classes below. Given the initial shell I provided for you. I provided a tester class called `RobotTester.java. This should be the output when running that file:`

```
=== Running Movement Program ===
2 3 EAST
=== Returning to Origin ===
=== Final Position Check ===
0 0 NORTH
```