**Java Assignment Detailed Requirement.**

Objective: Develop a Java application featuring a WalkingBoard class with fundamental and advanced functionalities. Implement board operations, player interactions, and ensure comprehensive testing using JUnit 5. Adhere to Java coding standards and submission guidelines.

**Base Exercise (9 points):**

1. **WalkingBoard Class:**
   * **Purpose:** Represents a board with a figure. Initialize with default or custom dimensions and values.
   * **Initialization:**
     + **Constructor 1:** Initializes a size×size board with default values (BASE\_TILE\_SCORE).
     + **Constructor 2:** Initializes using a 2D array with varying row lengths. Defaults applied where values are missing.
   * **Methods:**
     + getPosition(): Returns the figure's position [x, y].
     + getTile(int x, int y): Returns the value at (x, y); throws IllegalArgumentException for invalid positions.
     + getTiles(): Returns a copy of the board values.
     + getXStep() and getYStep(): Return changes in coordinates for movement.
     + moveAndSet(int newX, int newY, int newValue): Moves the figure and sets the value at the new position; cancels if out of bounds.
     + setAndMove(int newX, int newY, int newValue): Sets value at the new position before moving.
2. **JUnit 5 Tests:**
   * **testSimpleInit(size):** Validates board initialization with default values.
   * **testCustomInit(x, y, expected):** Checks custom initialization with specific dimensions and values.
   * **testMoves():** Tests various move scenarios, including edge cases.

**Extended Exercise (6 points):**

1. **Extended WalkingBoard Class:**
   * **Purpose:** Enhances the base class to support player interactions.
   * **Constructors:**
     + **Constructor 1:** Initializes players with at least two required. The first is a MadlyRotatingBuccaneer; others are simple Players.
   * **Player Movements:**
     + Players take turns moving, with the MadlyRotatingBuccaneer having special rotation rules based on turn count. Players' scores are updated based on board values.
2. **JUnit 5 Tests for Extended Functionality:**
   * **walk1 Test:** Validates board state and player scores with different player counts.
   * **walk2 Test:** Tests another board configuration to ensure correct movement and scoring.