ITN 261: Project 2 - Simple Game

Problem Statement:

You are tasked with writing a simple game, where a player goes from one place to another dodging obstacles. We will design and implement classes for a GameObject, Player, Obstacle, and display them on the screen. This is a two dimensional game where x and y correspond to player positions where the top left is 0,0 and x and y get larger as we go towards the bottom right.

Requirements:

- Design and implement a GameObject class with the following:
  - Data fields for x and y positions
- Design and implement a Player class with the following:
  - The Player class is a subclass of the GameObject class
  - User can enter a custom name
    - The name cannot be longer than 15 characters long
    - The name can only start with a letter (no numbers, special characters)
- Design and implement an Obstacle class with the following:
  - The Obstacle class is a subclass of the GameObject class
  - The obstacle can appear in any random location EXCEPT where the player is initially located (position 0,0)
- The sample runs are provided as examples; your output does not need to match exactly.
- Game.java is provided as starter code
  - ***You may NOT make any changes to the main method of the Game class.***

Grading Details:

- 75% Code quality/Correct usage of programming concepts
  - 45% Correct usage of classes (constructors, setters, getters, etc.)
  - 30% Correct usage of subclasses (inheritance)
- 5% Correct answer
- 5% Descriptive variable names
- 5% User friendliness (good prompts and outputs)
- 5% Comments (minimum required comments and human readable clarification for any unclear or unintuitive code)
- 5% Proper indentation/formatting

Starter Code:

```
	public static void main(String[] args) {

		// Player's constructor should initialize the Player
		// Initial location
		Player p1 = new Player();
```

```
            // Obstacle's constructor should initialize the obstacle
            // to a random location - which cannot overlap with p1 at 0, 0
            Obstacle obstacle = new Obstacle();

            // After the following statement, name will have a valid name
            // In other words, getValidName will have input validation
            String name = getValidName();
            p1.setName(name);

            // After the following statement, the screen should display
            // the player's name and position as well as the obstacle's position
            gameStatus(p1, obstacle);



      }
```

Sample runs (inputs highlighted in yellow):

```
Please enter a valid name (must start with a letter and be less than
15 characters):
MrSmith

Game status:
Player name: MrSmith
Display car positions:
P---------
----------
----O-----
----------
----------
```

Sample runs (inputs highlighted in yellow):

```
Please enter a valid name (must start with a letter and be less than
15 characters):
8weeksin
Please enter a valid name (must start with a letter and be less than
15 characters):
WhenWillThisSemesterEnd!
Please enter a valid name (must start with a letter and be less than
15 characters):
MrSmith
```

```
Game status:
Player name: MrSmith
Display car positions:
P---------
----------
--------O-
----------
----------
```