Joel Atkinson

September 14, 2025

Assignment 7.2

CSD402 Java for Programmers

UseFans Code:

//Joel Atkinson September 11, 2025 Assignment 7.2 CSD 402 Java for Programmers  
/\*The purpose of this assignment is to create a collection of fan instances that display without using the toString()  
method and use the "this" reference throughout the fan class. \*/  
  
import java.util.ArrayList;  
  
public class UseFans {  
 private ArrayList<Fan> fans = new ArrayList<>();  
  
 public void displayFans(ArrayList<Fan> fanList) {  
 for (Fan fan : fanList) {  
 System.*out*.println("Fan Details:");  
 System.*out*.println("Speed: " + fan.getSpeed());  
 System.*out*.println("Is On: " + fan.isFanOn());  
 System.*out*.println("Radius: " + fan.getRadius());  
 System.*out*.println("Color: " + fan.getColor());  
 System.*out*.println("-----------------");  
 }  
 }  
  
 public void displayFan(Fan fan) {  
 System.*out*.println("Single Fan Details:");  
 System.*out*.println("Speed: " + fan.getSpeed());  
 System.*out*.println("Is On: " + fan.isFanOn());  
 System.*out*.println("Radius: " + fan.getRadius());  
 System.*out*.println("Color: " + fan.getColor());  
 System.*out*.println("----------------");  
 }  
}

TestFan updated code for assignment 7.2:

//Joel Atkinson, September 11, 2025, CSD402 Assignment 6.2 Test Class  
/\*The purpose of this code is to test the output of the variables set up in the "Fan" class without using the toString()  
method. This is my updated TestFan code to follow assignment 7.2 instructions \*/  
  
import java.util.ArrayList;  
public class TestFan {  
 public static void main(String[] args) {  
 // Create an instance of UseFans  
 UseFans useFans = new UseFans();  
  
 // Create a collection of Fan instances  
 ArrayList<Fan> fanCollection = new ArrayList<>();  
 fanCollection.add(new Fan()); // Default fan  
 fanCollection.add(new Fan(3, true, 10, "blue")); // Custom fan  
 fanCollection.add(new Fan(1, false, 5, "red")); // Another custom fan  
  
 // Display the collection  
 System.*out*.println("Displaying all fans in the collection:");  
 useFans.displayFans(fanCollection);  
  
 // Display a single fan  
 System.*out*.println("Displaying a single fan:");  
 useFans.displayFan(fanCollection.get(0)); // Display the first fan  
  
 // Modify a fan and display again  
 Fan firstFan = fanCollection.get(0);  
 firstFan.setSpeed(2); // Change speed to MEDIUM  
 firstFan.setFanOn(true); // Turn it on  
 firstFan.setRadius(8); // Change radius  
 firstFan.setColor("green"); // Change color  
 System.*out*.println("Displaying modified first fan:");  
 useFans.displayFan(firstFan);  
 }  
}

Fan class code (unchanged from assignment 6.2):

//Joel Atkinson, September 2, 2025, CSD402 Assignment 6.2  
/\* The purpose of this assignment is to create a fan class with constants, mutable fields, setters and getters, as well  
as No-Argument and Argument constructors. Then build a test class to test the variables are output correctly \*/  
  
  
public class Fan {  
 // Constants for fan speed  
 public static final int *STOPPED* = 0;  
 public static final int *LOW* = 1;  
 public static final int *MEDIUM* = 2;  
 public static final int *FAST* = 3;  
  
 // Private mutable fields with defaults  
 private int speed = *STOPPED*;  
 private boolean fanOn = false;  
 private int radius = 6;  
 private String color = "white";  
  
 // Getter for speed  
 public int getSpeed() {return this.speed;  
 }  
  
 // Setter for speed  
 public void setSpeed(int newSpeed) {  
 if (newSpeed == *STOPPED* || newSpeed == *LOW* || newSpeed == *MEDIUM* || newSpeed == *FAST*) {  
 this.speed = newSpeed;  
 } else {  
 System.*out*.println("Invalid Speed! Please enter STOPPED, LOW, MEDIUM, or FAST");  
 }  
 }  
  
 // Getter for fanOn  
 public boolean isFanOn() {  
 return this.fanOn;  
 }  
  
 // Setter for fanOn  
 public void setFanOn(boolean newFanOn) {  
 this.fanOn = newFanOn;  
 }  
  
 // Getter for radius  
 public int getRadius() {  
 return this.radius;  
 }  
  
 // Setter for radius  
 public void setRadius(int newRadius) {  
 if (newRadius > 0) {  
 this.radius = newRadius;  
 } else {  
 System.*out*.println("Radius must be a positive int");  
 }  
 }  
  
 // Setter for color  
 public String getColor() {  
 return this.color;  
 }  
  
 // Getter for color  
 public void setColor(String newColor) {  
 this.color = newColor;  
 }  
  
 // No-argument constructor  
 public Fan() {  
 this.speed = *STOPPED*;  
 this.fanOn = false;  
 this.radius = 6;  
 this.color = "white";  
  
 }  
  
 // Argument constructor  
 public Fan(int speed, boolean fanOn, int radius, String color) {  
 this.speed = speed;  
 this.fanOn = fanOn;  
 if (radius> 0) {  
 this.radius = radius;  
 } else {  
 this.radius = 6;  
 System.*out*.println("Radius must be a positive, set to default of 6");  
 }  
 this.color = color;  
 }  
  
 // toString Method  
 public String toString() {  
 return "Fan: Speed " + speed + ", fanOn " + fanOn + ", radius " + radius + ", color " + color;  
 }  
  
}

Screenshot of updated TestFan code running without error:  
A screenshot of a computer

AI-generated content may be incorrect.