**Week 4 M6: Programming Assignment  
Fan Class**

Juan Macias Vasquez

Bellevue University

CSD402-H323 Java for Programmers (2261-DD)

**Jack Lusby**

September 7th, 2025

**Week 4 M6: Programming Assignment**

**Fan Class**

**GitHub Repository Link:**

<https://github.com/Juan551School/csd-402>

**Java Code**

//Juan Macias Vasquez

//Bellevue University

//CSD402-H323 Java for Programmers (2261-DD)

//Jack Lusby

//September 6th, 2025

**package** module6assignment;

**public** **class** FanClass {

// Constants

**public** **static** **final** **int** ***STOPPED*** = 0;

**public** **static** **final** **int** ***SLOW*** = 1;

**public** **static** **final** **int** ***MEDIUM*** = 2;

**public** **static** **final** **int** ***FAST*** = 3;

// Private fields

**private** **int** speed;

**private** **boolean** on;

**private** **double** radius;

**private** String color;

// No-argument constructor (default values)

**public** FanClass() {

**this**.speed = ***STOPPED***;//Default being stop

**this**.on = **false**;

**this**.radius = 6;//A private field named radius that holds the radius of the fan, default value of 6.

**this**.color = "white";//A String field that holds the color, default being white.

}

// Constructor with arguments

**public** FanClass(**int** speed, **boolean** on, **double** radius, String color) {

**this**.speed = speed;

**this**.on = on;

**this**.radius = radius;

**this**.color = color;

}

// Getters and Setters sectoin

**public** **int** getSpeed() {

**return** speed;

}

**public** **void** setSpeed(**int** speed) {

**this**.speed = speed;

}

**public** **boolean** isOn() {

**return** on;

}

**public** **void** setOn(**boolean** on) {

**this**.on = on;

}

**public** **double** getRadius() {

**return** radius;

}

**public** **void** setRadius(**double** radius) {

**this**.radius = radius;

}

**public** String getColor() {

**return** color;

}

**public** **void** setColor(String color) {

**this**.color = color;

}

// toString method

@Override

**public** String toString() {

**if** (on) {

**return** "Fan is ON\n" +

"Speed: " + speedToString(speed) + "\n" +

"Color: " + color + "\n" +

"Radius: " + radius;

} **else** {

**return** "Fan is OFF\n" +

"Color: " + color + "\n" +

"Radius: " + radius;

}

}

// Helper method to show speed as text

**private** String speedToString(**int** speed) {

**switch** (speed) {

**case** ***SLOW***: **return** "SLOW";

**case** ***MEDIUM***: **return** "MEDIUM";

**case** ***FAST***: **return** "FAST";

**default**: **return** "STOPPED";

}

}

// Test code (What the user ends up seeing)

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

// Fan using default constructor

FanClass fan1 = **new** FanClass();

System.***out***.println("Fan 1 (default constructor):");

System.***out***.println(fan1);

System.***out***.println();

// Fan using argument constructor

FanClass fan2 = **new** FanClass(FanClass.***FAST***, **true**, 10, "blue");

System.***out***.println("Fan 2 (argument constructor):");

System.***out***.println(fan2);

System.***out***.println();

// Demonstrating setters

fan1.setOn(**true**);

fan1.setSpeed(FanClass.***MEDIUM***);

fan1.setColor("red");

fan1.setRadius(8);

System.***out***.println("Fan 1 (after modifications):");

System.***out***.println(fan1);

}

}

**Pictures of Code Running**

