```
#include <Wire.h>

// ADXL345 I2C Address (most common)
#define A 0x53

// Configuration Constants
const int R = 200;            // Target Sample Rate: 200 Hz
const int D = 1000 / R;       // Delay: 5 milliseconds
const float S = 0.03125;      // Sensitivity for ±16g range (31.25 mg/LSB)

// Global Variables
int c = 0;
float g = 0;

// Function to Write to an ADXL345 Register
void w(byte r, byte v) {
  // CRITICAL FIX: The I2C commands MUST be inside the function body {}
  Wire.beginTransmission(A);
  Wire.write(r);
  Wire.write(v);
  Wire.endTransmission();
}

void setup() {
  Serial.begin(115200);
  Wire.begin();

  // 1. Set Data Rate to 200Hz (Register 0x2C, Value 0x0B)
  w(0x2C, 0x0B);

  // 2. Set Data Format to ±16g range (Register 0x31, Value 0x0F)
  // FIX for stuck values: This changes the range from 2g (0x0B in your old code) to 16g.
  w(0x31, 0x0F);

  // 3. Set Power Control Register to Measurement Mode (Register 0x2D, Value 0x08)
  w(0x2D, 0x08);

  Serial.println("c,g");
}

void loop() {
  Wire.beginTransmission(A);
  Wire.write(0x36); // Start reading from the Z-axis Least Significant Byte (L)
  Wire.endTransmission(false); // Do not release the bus (Repeated Start)
  Wire.requestFrom(A, 2); // Request 2 bytes (Z LSB and Z MSB)

  byte l = Wire.read(); // Read LSB (Low)
  byte h = Wire.read(); // Read MSB (High)
```

```
  // Combine bytes into a 16-bit signed integer
  int z = (int16_t)(((int)h << 8) | l);

  // Convert the raw 16-bit integer to 'g' (gravity) units
  g = (float)z * S;

  // Print results
  Serial.print(c);
  Serial.print(",");
  Serial.println(g, 4);

  // CRITICAL FIX: Counter increment and delay MUST be inside loop()
  c++;
  delay(D);
}
```