**Question 3 (20 marks)**

A gray value is a numeric value that represents a color that is a shade of gray. There are two common ways to represent a gray value:

1. as an integer value between 0 and 255 (inclusive) where 0 corresponds to the color black, 255 corresponds to the color white, and values in between correspond to some shade of gray
2. as a floating-point value between 0.0 and 1.0 (inclusive) where 0.0 corresponds to the color black, 1.0 corresponds to the color white, and values in between correspond to some shade of gray

The formulas for converting between integer gray values, $I$, and floating-point gray values, $F$, are:

$$F = \frac{I}{255}$$

$$I = round(255 \times F)$$

Implement a class called `Gray` that can represent gray values in both integer and floating-point forms.

Your class must provide the following features:

- a constructor that initializes the gray value to some caller specified integer value
  - the constructor should validate its arguments if needed
- a static factory method that returns a `Gray` object having some caller specified floating-point value
  - the constructor should validate its arguments if needed
- a non-static method that returns the value of a `Gray` object as a floating-point value
- an overridden `equals` method
  - two `Gray` objects are equal if and only if they have the same numeric value and the same representation (integer or floating-point)
    - for example, the gray values 51 and 0.2 are not equal (even though they represent equivalent gray values) because one value is in integer form and the other value is in floating-point from
- an overridden `hashCode` method that satisfies the general contract of the `hashCode` method
- an overridden `toString` method that returns a string containing the numeric value of the gray value and its representation (integer or floating-point)