1. Compare and contrast the float and Decimal classes' benefits and drawbacks.

Ans: Float and decimal types store numerical values. Float is a single precision (32 bit) floating point data type and decimal is a 128-bit floating point data type. Decimal gives a high degree of accuracy and easy to avoid rounding errors whereas Float is used when you store scientific numbers and for better performance. Performance of Decimals is slower than float data types.

2. Decimal('1.200') and Decimal('1.2') are two objects to consider. In what sense are these the same object? Are these just two ways of representing the exact same value, or do they correspond to different internal states ?

Ans: Both values are same but internal representation at storage is different. Precision differs, Decimal('1.200') gives internally 1.200 and Decimal('1.2') gives 1.2.

3. What happens if the equality of Decimal('1.200') and Decimal('1.2') is checked ?

Ans: Both values are checked to be equal , they only differ in precision.

```
In [7]:  1 from decimal import Decimal
         2 Decimal('1.200') == Decimal('1.2')

Out[7]: True
```

4. Why is it preferable to start a Decimal object with a string rather than a floating-point value ?

Ans: Decimal can store float value with absolute precision. But when float value is given as a Decimal object,it first has to be converted from floating point value which might already have a rounding error.

5. In an arithmetic phrase, how simple is it to combine Decimal objects with integers ?

Ans: It can simply be done using the decimal().

6. Can Decimal objects and floating-point values be combined easily ?

Ans: Decimal objects can not be combined with floating-point value as it generates a type error. To perform operations, the floating point has to be converted to a Decimal.

7. Using the Fraction class but not the Decimal class, give an example of a quantity that can be expressed with absolute precision ?

Ans: Value of 0.25 will be represented as ¼ .

8. Describe a quantity that can be accurately expressed by the Decimal or Fraction classes but not by a floating-point value.

Ans: In decimal floating point, 0.1 + 0.1 + 0.1 - 0.3 is exactly equal to zero. In binary floating point, the result is 5.551115231257827e-017

9. Consider the following two fraction objects: Fraction(1, 2) and Fraction(1, 2). (5, 10). Is the internal state of these two objects the same? Why do you think that is ?

Ans: Yes, they are the same as they reduce to ½.

10. How do the Fraction class and the integer type (int) relate to each other? Containment or inheritance ?

Ans: Fraction class and integer type(int) are related by a container.It contains two ints, one the numerator and the other the denominator