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

**Ans:** Use floats when convenience and speed matter. A float gives you an approximation of the number you declare. For example, if I print 0.1 with 18 decimals places, I don’t actually get 0.1 but instead an approximation.

>>> print(f"{0.1:.18f}")

0.100000000000000006

>>> .1 + .1 + .1 == .3

False

>>> .1 + .1 + .1

0.30000000000000004

Use decimals when precision matters, such as with financial calculations. Decimals are more precise than floats. The precision benefits of a decimal outweigh the performance benefits of a float.

Let’s look at the previous examples with decimals instead of floats.

>>> from decimal import Decimal

>>> print(f"{Decimal('0.1'):.18f}")

0.100000000000000000

>>> Decimal('.1') + Decimal('.1') + Decimal('.1') == Decimal('.3')

True

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:** these are same objects since both objects are of decimal class, also yes these are just two ways to representing the exact same value.

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

**Ans:** It will return true since both are same.

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

**Ans:** Because**,** Decimals are more precise than floats.

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

**Ans:** its very simple, nothing special nee to do, just use integers with decimals and the value which will return of type decimal.

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

**Ans:** We cannot combine float with decimal directly, to combine we need to convert float into 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:**

from fractions import Fraction

print (Fraction(18, 5) / Fraction(18, 10))

it will return 2.

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

**Ans:**

from decimal import Decimal

print (1/10)

print(f"{0.1:.18f}")

print(f"{Decimal('0.1'):.18f}")

Q9.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 internal state of these two objects are the same, because both are equal values.

Q10. How do the Fraction class and the integer type (int) relate to each other? Containment or inheritance?

**Ans:** Fractions and int type are related to each other by containment concept, because fractions contains int type in itself, numerator and denominator of a fraction is itself are integers.