Q1. Compare and contrast the float and Decimal classes' benefits and drawbacks.

- Float: Benefits - Efficiency, broad support, easy to use. Drawbacks - Limited precision, rounding errors, not suitable for precise financial calculations.

- Decimal: Benefits - High precision, exact representation of decimal numbers, suitable for financial calculations. Drawbacks - Less efficient, requires more memory.

Q2. Are Decimal('1.200') and Decimal('1.2') the same object or different internal states?

- Answer: They are different objects with different internal states. The trailing zeros in the string representation do not affect the internal representation of the Decimal object.

Q3. What happens when equality of Decimal('1.200') and Decimal('1.2') is checked?

- Answer: They are considered equal because the Decimal class handles trailing zeros and compares the numerical values.

Q4. Why is it preferable to start a Decimal object with a string rather than a floating-point value?

- Answer: Starting with a string ensures the exact representation of decimal values, avoiding potential precision issues and rounding errors associated with floating-point values.

Q5. How simple is it to combine Decimal objects with integers in an arithmetic phrase?

- Answer: Combining Decimal objects with integers is straightforward in arithmetic operations; they can be used together seamlessly.

Q6. Can Decimal objects and floating-point values be combined easily?

- Answer: While you can combine Decimal objects and floating-point values in operations, be cautious of potential precision issues. It's safer to use Decimal objects consistently when precision matters.

Q7. Using the Fraction class, provide an example of a quantity that can be expressed with absolute precision but not by floating-point values.

- Answer: An example is `Fraction(1, 3)` representing one-third. In floating-point, this cannot be represented with absolute precision due to binary approximations.

Q8. Describe a quantity accurately expressible by the Decimal or Fraction classes but not by a floating-point value.

- Answer: A repeating decimal like `1/3` (0.3333...) can be accurately expressed using Decimal or Fraction, but floating-point representations will involve rounding and lack precision.

Q9. Are Fraction(1, 2) and Fraction(1, 2, 5, 10) the same? Why?

- Answer: Yes, they are the same. The `Fraction` class simplifies fractions to their simplest form, so both represent one-half.

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

- Answer: The Fraction class does not inherit from the integer type (int) but can be constructed from integers. There is a containment relationship, as Fraction objects can hold integer values and represent them as fractions with a numerator and denominator.