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

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?

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

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

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

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

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

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

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?

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

Answer:

1. The float and Decimal classes are both used for representing numerical values, but they differ in their precision and rounding behavior. Floats are implemented in hardware and have a limited precision, which can lead to rounding errors when performing arithmetic operations. Decimal, on the other hand, uses a base-10 system and can represent numbers with arbitrary precision. This makes Decimal suitable for financial calculations and other applications where exact results are required. However, Decimal is slower and uses more memory than floats.
2. Decimal('1.200') and Decimal('1.2') are different objects that represent the same value. They correspond to different internal states because the first one has a trailing zero and the second one does not.
3. The equality of Decimal('1.200') and Decimal('1.2') is True, because they represent the same value.
4. It is preferable to start a Decimal object with a string rather than a floating-point value because the latter can introduce rounding errors. For example, Decimal(0.1) is not exactly equal to 0.1, whereas Decimal('0.1') is.
5. It is simple to combine Decimal objects with integers in an arithmetic phrase, because Decimal supports the standard arithmetic operators.
6. Decimal objects and floating-point values can be combined, but the result will be a Decimal object with the precision of the most precise operand.
7. An example of a quantity that can be expressed with absolute precision using the Fraction class is 1/3, which cannot be represented exactly as a decimal.
8. A quantity that can be accurately expressed by the Decimal or Fraction classes but not by a floating-point value is 1/10, which is a repeating decimal in base 2.
9. The internal state of Fraction(1, 2) and Fraction(1, 2, 5, 10) is different, because the latter is in reduced form. However, they represent the same value and are equal.
10. The Fraction class and the integer type (int) are related by containment, because a Fraction object can contain an integer value as its numerator or denominator. However, they are not related by inheritance.