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

Ans: decimals are more precise than floats.

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?

Ans: 1.20 is exactly the same as 1.2

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

Ans: The precision of a decimal number is higher than both float and double.

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

Ans: a positive or negative whole number with a decimal point.

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

Ans: A float is a floating-point number, which means it is a number that has a decimal place.

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.

Ans: Floating point numbers or floats are another Python data type. Floats are decimals, positive, negative and zero. Floats can also be numbers in scientific notation which contain exponents. In Python, a float can be defined using a decimal point .

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?