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

ANS:

Floating point data type represent number values with fractional parts. Decimal accurately represent any number within the precision of the decimal format, whereas Float cannot accurately represent all numbers.

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:

Using ceil() function  
  
Use the ceil() function(returns a ceiling value of the number i.e., the smallest integer greater than or equal to the number), to round the number upto the 2 decimal places and print the resultant number.

In Python, there is a module called Decimal, which is used to do some decimal floating point related tasks. This module provides correctly-rounded floating point arithmetic.

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

ANS:

It depends on the person who does the measurement.

If the person knows about this stuff, they will only write 1.20 if they are confident about their ability to measure to the second decimal place.

However, someone else might have a tool that can't measure more digits after the 2. Still it is possible that they might write 1.20. So we can never be sure.

But if we are the one doing the measurement, only write 1.20 if you are sure that it is not 1.21 or something like that.

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

ANS:

The [decimal](https://docs.python.org/3/library/decimal.html#module-decimal) module provides support for fast correctly rounded decimal floating point arithmetic. It offers several advantages over the [float](https://docs.python.org/3/library/functions.html#float) datatype:

* Decimal “is based on a floating-point model which was designed with people in mind, and necessarily has a paramount guiding principle – computers must provide an arithmetic that works in the same way as the arithmetic that people learn at school.” – excerpt from the decimal arithmetic specification.
* Decimal numbers can be represented exactly. In contrast, numbers like 1.1 and 2.2 do not have exact representations in binary floating point. End users typically would not expect 1.1 + 2.2 to display as 3.3000000000000003 as it does with binary floating point.
* The exactness carries over into arithmetic. In decimal floating point, 0.1 + 0.1 + 0.1 - 0.3 is exactly equal to zero. In binary floating point, the result is 5.5511151231257827e-017. While near to zero, the differences prevent reliable equality testing and differences can accumulate. For this reason, decimal is preferred in accounting applications which have strict equality invariants.

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

ANS:

The [decimal](https://docs.python.org/3/library/decimal.html#module-decimal) module provides support for fast correctly rounded decimal floating point arithmetic. It offers several advantages over the [float](https://docs.python.org/3/library/functions.html#float) datatype:

* Decimal “is based on a floating-point model which was designed with people in mind, and necessarily has a paramount guiding principle – computers must provide an arithmetic that works in the same way as the arithmetic that people learn at school.” – excerpt from the decimal arithmetic specification.

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

ANS:

Decimal objects cannot generally be combined with floats or instances of fractions.Fraction in arithmetic operations: an attempt to add a Decimal to a float , for example, will raise a TypeError .

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

ANS:

Fixed-Point Representation −  
  
This representation has fixed number of bits for integer part and for fractional part. For example, if given fixed-point representation is IIII. FFFF, then you can store minimum value is 0000.0001 and maximum value is 9999.9999.

Decimal Fixed-Point Representation  
  
Fixed-point representation has a radix point known as decimal point. Fixed-point numbers having decimal points at the right end of the number are treated as integers because the fixed-point numbers having decimal points at the left end of the number are treated as fractions.

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

ANS:

A decimal floating-point constant is a mantissa followed by an exponent. The mantissa is a decimal fixed-point constant. The exponent is the letter E, S, D, or Q followed by an optionally-signed decimal integer of four or less digits (meaning 10 to the power of this integer).

The precision of floating-point numbers is either single or double, based on the number of hexadecimal digits in the fraction. A small integer is a binary integer with a precision of 15 bits. The range of small integers is -32768 to +32767. A large integer is a binary integer with a precision of 31 bits.

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:

Fractions are the terms used to determine the parts of a whole object. For example, a pizza is divided into four pieces, so each piece of it is represented as 1/4th of the pizza.

In real life, we will many examples of fractions, such as: If a pizza is divided into two equal parts, then each part is equal to half of the whole pizza. If we divide a slice of watermelon into three equal parts then each part is equal 1/3rd of the whole.

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

ANS:

Inheritance is a parent-child relationship where we create a new class by using existing class code. It is just like saying that “A is type of B”. For example is “Apple is a fruit”, “Ferrari is a car”.

Inheritance models the is-a relationship between two classes. A strong is-a relationship describes a direct inheritance relationship between two classes. A weak is-a relationship describes that a class has certain properties. A strong is-a relationship can be represented using class inheritance.