```
#TheFractionClass.py
""" Illustrate a class that supports various operations
of
fractions. Shows how to overload "+" and "*"
"""

def
gcd(p,q):
    """ Returns the greatest common divisor of p and q.


PreC: p and q are integers and q is nonzero.
    """
    a = abs(p)
    b =
abs(q)
    r = a%b
    while r>0:
        a = b
        b = r
        r = a%b
    return b


class Fraction:
    """
    A class that supports operations with fractions.


    Attributes:
        num: the numerator [int]
        den: the denominator [int>0]


        num and den are reduced to lowest terms, that is,
        one is their greatest
common divisor.

    """
    def __init__(self,p,q=1):

""" Returns a reference to a Fraction Object that represents p/q


 PreC p and q are ints and q is nonzero
        """
        # Reduce to lowest
terms...
        d = gcd(p,q)
        self.num = p/d
        self.den = q/d

    def
__str__(self):
        """ Pretty prints self
        """

return  '%1d/%1d' % (self.num,self.den)

    def __add__(self,f):

""" Returns a Fraction that is the sum of self and f.
        If f1 is a
Fraction and f2 is a Fraction or an int, then
        f3 = f1+f2 is a Fraction object that
represents their sum.

        PreC: f is either an int or a Fraction

"""

        if isinstance(f,Fraction):
            # f is a fraction
```

```
     N = self.num*f.den + self.den*f.num
            D = self.den*f.den
        else:

  # f is an int
            N = self.num + self.den*f
            D = self.den
        return
Fraction(N,D)

    def __mul__(self,f):
        """  Returns a Fraction that
is the product of self and f.
        If f1 is a Fraction and f2 is a Fraction or an int, then

       f3 = f1+f2 is a Fraction object that represents their product.



Returns a fraction that is the product of self and f
        PreC: f is either an int or a
fraction
        """
        if isinstance(f,Fraction):
            # f is a
Fraction
            N = self.num*f.num
            D = self.den*f.den
        else:

 # f is an int
            N = self.num*f
            D = self.den
        return
Fraction(N,D)

    def __eq__(self,f):
        """ Returns True if and
only if  self represents the same fraction.
        as f.

        PreC: f is a
Fraction
        """
        return (self.num == f.num) and (self.den == f.den)


    def negate(self):
        """ Returns the negative of self.

"""
        return Fraction(-self.num,self.den)

    def invert(self):

  """ Returns the reciprocal of self
        PreC: self is not zero

"""
        return Fraction(self.den,self.num)
```