```
"""
The purpose of this program is to create a truth table that will demonstrate that
De Morgan's Law is a tautology.  A lexicographical truth table will be constructed
for the input statements, their negations, and the "and" and "or" of those statements.

I was able to remove a lot of unnecessary code from the previous two assignments
by using native Python.

The iff function performs the if and only if calculation and returns a boolean
that represents the result.  Results are cast to strings before printing to easily
add a trailing space to the true answers for the sake of table layout.
"""

class deMorgan:
    def __init__(self):

        #Global strings
        self.header = "|  P   |  Q   | P^Q |~ P^Q| ~P   | ~Q   |~Pv~Q| iff |"
        self.line = "+-----+-----+-----+-----+-----+-----+-----+-----+"

        #Display the header for the truth table
        print ("\n")
        print (self.line)
        print (self.header)
        print (self.line)

        #Arrays of P and Q values
        p = [True, True, False, False]
        q = [True, False, True, False]

        #Iterate over the values and call printTruth on each combination
        for i in range(0, 4):
            self.printTruth(p[i], q[i])
            print (self.line)

    #Performs the if and only if comparison: (iff)
    def iff(self, p, q):
        if ((not (p and q)) and ((not p) or (not q))):
            return True
        elif (not (not (p and q)) and (not (not p) or (not q))):
            return True
        else:
            return False

    #Prints the line in the truth table
    def printTruth(self, p, q):

        print ("|" + self.s(p) +                            #(P)
            "|" + self.s(q) +                               #(Q)
            "|" + self.s(p and q) +                         #(P^Q)
            "|" + self.s(not (p and q)) +                   #~(P^Q)
            "|" + self.s(not p) +                           #~(P)
            "|" + self.s(not q) +                           #~(Q)
            "|" + self.s((not p) or (not q)) +              #~(P)v~(Q)
```

```python
            "|" + self.s(self.iff(p,q)) +           #iff
            "|")

    #Add spaces to true to keep entries even
    def s(self, p):
        while p == True:
            p = str(p) + " "

        p = str(p)
        return p

d = deMorgan()
```