```
'''
Created on 14 Oct 2014
@author: bob
'''
class Function(object):
    '''
    A class to represent function objects, these functions
    must be able to be evaluated and to be derived at some point.
    '''
    def __init__(self, delta=0.001):
        '''
        Constructor
        '''
        self._DELTA = delta
        pass
    def evaluate(self,x):
        '''
        A method to evaluate the function
        '''
        raise NotImplementedError
    def derivative(self,x):
        '''
        A method to calculate the derivative of the function
        '''
        return (self.evaluate(x+self._DELTA/2) - self.evaluate(x-self._DEL
class P(Function):
    def __init__(self,Ksqr,sigma,g,wsqr):
        Function.__init__(self)
        self.Ksqr = Ksqr
        self.sigma = sigma
        self.g = g
        self.wsqr = wsqr
    def evaluate(self, x):
        return self.wsqr*(rho0(self.Ksqr,self.sigma,self.g,self.wsqr).eval

class Q(Function):
    def __init__(self,Ksqr,sigma,g,wsqr):
        Function.__init__(self)
        self.Ksqr = Ksqr
        self.sigma = sigma
        self.g = g
        self.wsqr = wsqr
    def evaluate(self, x):
        return self.Ksqr*((rho0(self.Ksqr,self.sigma,self.g,self.wsqr).eva
                           (rho0(self.Ksqr,self.sigma,self.g,self.wsqr).de

class rho0(Function):
    def __init__(self,Ksqr,sigma,g,wsqr):
```

```
49          Function.__init__(self)
50          self.Ksqr = Ksqr
51          self.sigma = sigma
52          self.g = g
53          self.wsqr = wsqr
54      def evaluate(self, x):
55          return (1+self.sigma*x)
```