In [1]:
```
from sympy import Symbol, Derivative, symbols
x= Symbol('x')
y= Symbol('y')
function = x**2 + y**2 + 10
partialderiv= Derivative(function, y)
dfy = partialderiv.doit()
partialderiv= Derivative(function, x)
dfx = partialderiv.doit()

print(f'first order derivatives of given function are df/dx = {dfx} ,df/dy = {dfy

#step 1 initialise values
x1 = -1.0
y1 = 2.0  #variable x, y
itr = 100  #epoches
learning_rate = 0.1  #learning rate
for i in range(0,itr):
    #step 2 substitute x ,y in first order derivatives df/dx, df/dy
    x = symbols('x')
    y = symbols('y')
    dfxv = dfx.subs(x, x1)
    dfxv = round(dfxv,2)
    dfyv = dfy.subs(y, y1)
    dfyv = round(dfyv,2)
    #step3 find change in x , y
    dx = (-1.0)*learning_rate*dfxv
    dy = (-1.0)*learning_rate*dfyv
    #step4 update variable
    x1 = x1 + dx
    x1 = round(x1, 2)
    y1 = y1 + dy
    y1 = round(y1, 2)
    #step5 increment iterations
    #step6 break loop if iterations exceed no of epoches
#step7 print variable x1, y1
print(x1, y1)
print(f'minimum value obtained at x = {x1} ,y ={y1} for given function')
```

```
first order derivatives of given function are df/dx = 2*x ,df/dy = 2*y
-0.02 0.02
minimum value obtained at x = -0.02 ,y =0.02 for given function
```

In [ ]:

In [ ]:

In [ ]: