In [46]:

```
# This Linear Regression model for Stock Market Predecting

import pandas_datareader.data as web # import pandas_datareader to read data from the web - stock
market
import datetime as dt
import pandas as pd
import matplotlib.pyplot as plt  # for ploting the graph
from sklearn import linear_model  # linear regression in built math function - y= mx+ b
```

In [39]:

```
start = dt.datetime(2018,11,17)
end = dt.datetime.today()
stock = 'DHFL.NS'
df = web.DataReader(stock,'yahoo',start, end) # reading the data from web - yahoo fin
print(df.head())
```

```
                 High         Low        Open       Close    Volume  \
Date
2018-11-19  235.000000  225.699997  227.000000  231.399994   9529435
2018-11-20  241.449997  220.000000  229.649994  225.300003  22120570
2018-11-21  238.800003  224.250000  227.699997  234.850006  24463945
2018-11-22  242.850006  230.000000  238.649994  232.750000  20766936
2018-11-26  234.899994  218.600006  234.899994  222.399994  12876386

             Adj Close
Date
2018-11-19  231.399994
2018-11-20  225.300003
2018-11-21  234.850006
2018-11-22  232.750000
2018-11-26  222.399994
```

In [27]:

```
df = df.rename(columns = {'Adj Close':'CLOSE'})    # rename the Adj Close to Close
print(df.tail())
```

```
                 High         Low        Open       Close    Volume  \
Date
2019-02-12  108.750000  103.500000  104.400002  106.449997  12866364
2019-02-13  112.650002  106.300003  108.000000  111.199997  14431475
2019-02-14  131.350006  103.949997  104.000000  127.949997  54153006
2019-02-15  129.449997  121.000000  128.100006  123.199997  19036558
2019-02-18  135.500000  124.900002  124.900002  128.850006  32069129

                CLOSE
Date
2019-02-12  106.449997
2019-02-13  111.199997
2019-02-14  127.949997
2019-02-15  123.199997
2019-02-18  128.850006
```

In [28]:

```
data_source = r'E:\Data_Set\DHFL.xlsx'    # writing the web data into excel
df.to_excel(data_source)
df = pd.read_excel(data_source)        # reading the dataframe
```

In [29]:

```
print(df.tail())
```

```
         Date        High         Low        Open       Close    Volume  \
59 2019-02-12  108.750000  103.500000  104.400002  106.449997  12866364
```

```
60 2019-02-13  112.650002  106.300003  108.000000  111.199997  14431475
61 2019-02-14  131.350006  103.949997  104.000000  127.949997  54153006
62 2019-02-15  129.449997  121.000000  128.100006  123.199997  19036558
63 2019-02-18  135.500000  124.900002  124.900002  128.850006  32069129

         CLOSE
59  106.449997
60  111.199997
61  127.949997
62  123.199997
63  128.850006
```

In [30]:

```
df = pd.read_excel(data_source, index_col = 'Date')      # changing the data to index coloum
print(df.head())
```

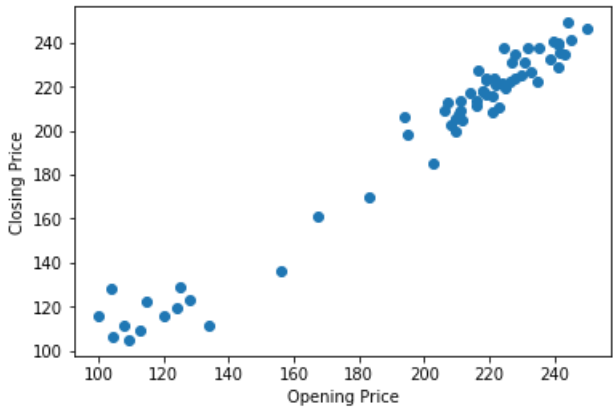
```
                 High         Low        Open       Close     Volume  \
Date
2018-11-19  235.000000  225.699997  227.000000  231.399994    9529435
2018-11-20  241.449997  220.000000  229.649994  225.300003  22120570
2018-11-21  238.800003  224.250000  227.699997  234.850006  24463945
2018-11-22  242.850006  230.000000  238.649994  232.750000  20766936
2018-11-26  234.899994  218.600006  234.899994  222.399994  12876386

                CLOSE
Date
2018-11-19  231.399994
2018-11-20  225.300003
2018-11-21  234.850006
2018-11-22  232.750000
2018-11-26  222.399994
```

In [31]:

```
%matplotlib inline                # ploting the graph
plt.xlabel('Opening Price')
plt.ylabel('Closing Price')
plt.scatter(df.Open, df.CLOSE)        # open: independent variable  close: depedent variable
```

Out[31]:

```
<matplotlib.collections.PathCollection at 0x26e1ba08748>
```




In [32]:

```
reg = linear_model.LinearRegression()       # calling linear regression
reg.fit(df[['Open']], df.CLOSE              # passing the variables to the model  independent,  dep
endent
```

Out[32]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [33]:

```
today_open_price = 124.900002                  # is the value of x in the formula

predict_today_close_price = reg.predict([[today_open_price]])    #  predicting the value
```

In [42]:

```
reg.intercept_                     # is the value of b in the formula
```

Out[42]:

```
3.285443530585411
```

In [43]:

```
reg.coef_                         # is the value of m in the formula
```

Out[43]:

```
array([0.97596752])
```

In [44]:

```
print('Predicted todays close price:' , predict_today_close_price) # predicted value  is y in the formula
```

```
Predicted todays close price: [125.18378854]
```

In [47]:

```
(0.97596752 * 124.900002) + 3.285443530585411        # y = m x + b
```

Out[47]:

```
125.18378873052045
```

In [ ]:

```
# This linear Regression Model is only a example for Bigners
```