In [1]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Loading CSV data
data = pd.read_csv("Salary.csv")
exp = data["YearsExperience"]
salary = data["Salary"]
data.head()
```

Out[1]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

In [2]:

```
data.describe()
```

Out[2]:

	YearsExperience	Salary
count	35.000000	35.000000
mean	6.308571	83945.600000
std	3.618610	32162.673003
min	1.100000	37731.000000
25%	3.450000	57019.000000
50%	5.300000	81363.000000
75%	9.250000	113223.500000
max	13.500000	139465.000000

In [3]:

```
ax = data.plot(x="YearsExperience", y="Salary")
```

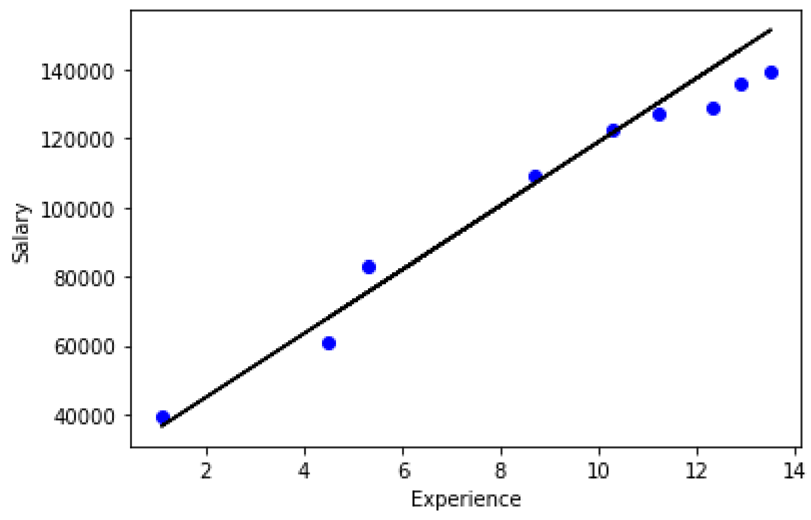
In [4]:

```
X = np.array(exp).reshape(-1, 1)
y = np.array(salary).reshape(-1, 1)

# Splitting data for test and train
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)

# Initializing LinearRegression object
regr = LinearRegression()

# Fitting Regression Line with train data
regr.fit(X_train, y_train)

# Printing the Accuracy Score with test data
print(f"Accuracy Score: {(regr.score(X_test, y_test) * 100):.2f}%")


# Predicting the values with test data
y_pred = regr.predict(X_test)

# Plotting the Regression Line
fig, ax = plt.subplots()
ax.scatter(X_test, y_test, color ='b')
ax.plot(X_test, y_pred, color ='k')
ax.set_xlabel("Experience")
ax.set_ylabel("Salary")
# plt.show()

# Converting the predicted 2D array to 1D array
pre = [i[0] for i in y_pred]

# Creating Dataframe Table to show the difference
pred = pd.DataFrame({
"Actual Value": [i[0] for i in y_test],
"Predicted": pre
})

# Predicted Value
pred
```

```
Accuracy Score: 95.03%
```

Out[4]:

	Actual Value	Predicted
0	127345	130119.621464
1	128765	140311.305711
2	135675	145870.406209
3	122391	121780.970716
4	61111	68042.999233
5	109431	106956.702721
6	39343	36541.429744
7	139465	151429.506707
8	83088	75455.133231



Loss Formula:

$$Loss[f] = \frac{1}{n}\sum_{i=1}^{n}(f(x^i) - y^i) = Squaredloss$$

where $Loss[f]$ is the model, $f(x^i)$ is a predicted value, y^i is a actual value, x^i or y^i is a n^{th} vector.