In [2]:
```
#exp:7
```

In [3]:
```
#Name:Vedant M.Padole
#Roll no:42
#Sec:C
#Subject:ET1
#Date:
```

In [4]:
```
import pandas as pd
```

In [6]:
```
import os
```

In [7]:
```
os.getcwd()
```

Out[7]:
```
'C:\\Users\\DELL'
```

In [8]:
```
os.chdir('C:\\Users\\DELL\\Desktop')
```

In [9]:
```
df=pd.read_csv("Salary_Data.csv")
```

In [10]:
```
df.head()
```

Out[10]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

In [11]:
```
df.tail()
```

Out[11]:

	YearsExperience	Salary
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [12]:
```
df.shape
```

Out[12]:
```
(30, 2)
```

In [13]:
```
df.size
```

Out[13]:
```
60
```

In [14]:
```
df.ndim
```

Out[14]: 2

In [15]:
```
df.describe
```

Out[15]:
```
<bound method NDFrame.describe of     YearsExperience    Salary
0               1.1    39343.0
1               1.3    46205.0
2               1.5    37731.0
3               2.0    43525.0
4               2.2    39891.0
5               2.9    56642.0
6               3.0    60150.0
7               3.2    54445.0
8               3.2    64445.0
9               3.7    57189.0
10              3.9    63218.0
11              4.0    55794.0
12              4.0    56957.0
13              4.1    57081.0
14              4.5    61111.0
15              4.9    67938.0
16              5.1    66029.0
17              5.3    83088.0
18              5.9    81363.0
19              6.0    93940.0
20              6.8    91738.0
21              7.1    98273.0
22                7.9 101302.0
23                8.2 113812.0
24                8.7 109431.0
25                9.0 105582.0
26                9.5 116969.0
27                9.6 112635.0
28               10.3 122391.0
29               10.5 121872.0>
```

In [16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0    YearsExperience 30 non-null     float64
 1   Salary           30 non-null     float64
dtypes: float64(2)
memory usage: 608.0 bytes
```

In [17]:
```
df.isna()
```

Out[17]:

	YearsExperience	Salar y
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False

	YearsExperience	Salar y
9	False	False
10	False	False
11	False	False
12	False	False
13	False	False
14	False	False
15	False	False
16	False	False
17	False	False
18	False	False
19	False	False
20	False	False
21	False	False
22	False	False
23	False	False
24	False	False
25	False	False
26	False	False
27	False	False
28	False	False
29	False	False

In [18]:

```
df.isna().any()
```

Out[18]:

```
YearsExperience    False
Salary             False
dtype: bool
```

In [19]:

```
df.isna().sum()
```

Out[19]:

```
YearsExperience    0
Salary             0
dtype: int64
```

In [20]:

```
x=df.drop('Salary',axis=1)
```

In [21]:

```
x.head()
```

Out[21]:

	YearsExperience
0	1.1
1	1.3
2	1.5
3	2.0

	YearsExperience
4	2.2

In [22]:
```
y=df.Salary
```

In [23]:
```
y.head()
```

Out[23]:
```
0    39343.0
1    46205.0
2    37731.0
3    43525.0
4    39891.0
Name: Salary, dtype: float64
```

In [24]:
```
from sklearn.model_selection import train_test_split
```

In [25]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_stat
```

In [26]:
```
print(x_train.shape)
```

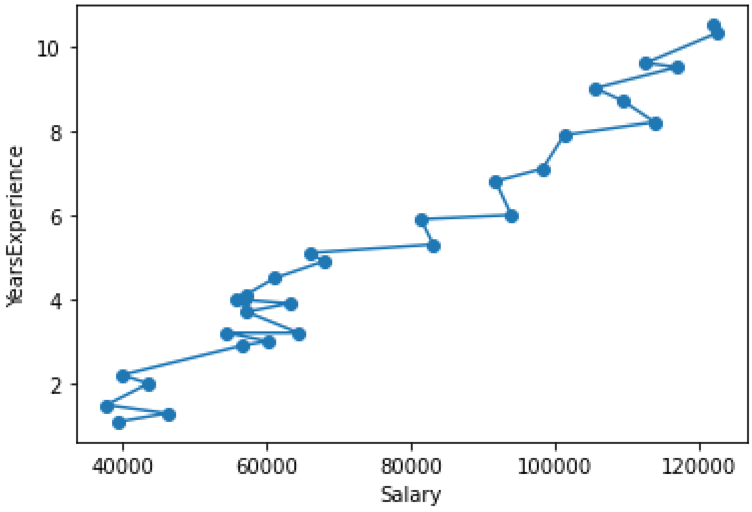
```
(24, 1)
```

In [27]:
```
x_test.shape
```

Out[27]:
```
(6, 1)
```

In [28]:
```
import matplotlib.pyplot as plt
```

In [29]:
```
plt.plot(df['Salary'], df['YearsExperience'], marker='o')
plt.xlabel("Salary")
plt.ylabel("YearsExperience")
plt.show()
```



Model Fitting

In [32]:
```
from sklearn.linear_model import LinearRegression
LR=LinearRegression()
LR.fit(x_train,y_train)
```

Out[32]:
```
LinearRegression()
```

```
In [33]: #Assigning Coefficient (Slope) to m
         m=LR.coef_
```

```
In [34]: print("Coefficient :" , m)
```

```
Coefficient : [9312.57512673]
```

```
In [35]: #Assigning Y-intercept to a
         c=LR.intercept_
```

```
In [36]: print("Intercept : ", c)
```

```
Intercept : 26780.099150628186
```

Evaluation Metrics

```
In [37]: from sklearn import metrics
```

```
In [38]: Accuracy = LR.score(x_test, y_test)
         Accuracy
```

```
Out[38]: 0.988169515729126
```

Conclusion :

Thesimple linearregression model shows a clear linear relationship between the variables, allowing us to predict outcomes based on this trend with reasonable accuracy

```
In [ ]:
```