Practical 1

Aim: Implementing a simple linear regression model

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
url='http://bit.ly/w-data'
s_data=pd.read_csv(url)
```

In [2]:
```
s_data.shape
```

Out[2]:
```
(25, 2)
```

In [3]:
```
s_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

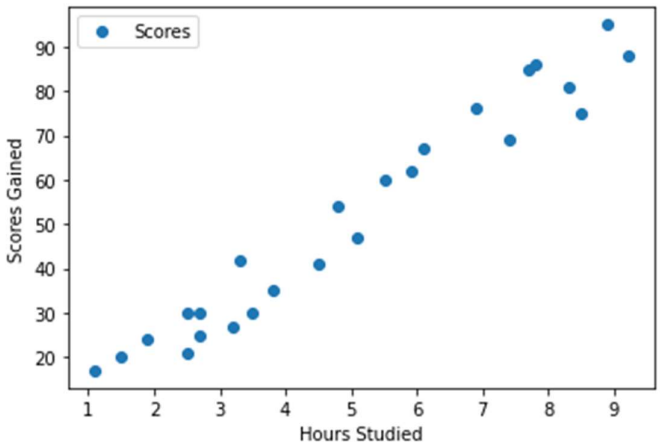
In [4]:
```
s_data.sample(4)
```

Out[4]:

	Hours	Scores
4	3.5	30
1	5.1	47
9	2.7	25
7	5.5	60

In [5]:
```
s_data.plot(x='Hours',y='Scores',style='o')
plt.xlabel('Hours Studied')
plt.ylabel('Scores Gained')
```

Out[5]:
```
Text(0, 0.5, 'Scores Gained')
```



In [9]:
```
X=s_data.iloc[:,:-1].values
y=s_data.iloc[:,-1].values
X
```

Out[9]:
```
array([[2.5],
       [5.1],
       [3.2],
       [8.5],
       [3.5],
       [1.5],
       [9.2],
       [5.5],
       [8.3],
       [2.7],
       [7.7],
       [5.9],
       [4.5],
       [3.3],
       [1.1],
       [8.9],
       [2.5],
       [1.9],
       [6.1],
       [7.4],
       [2.7],
       [4.8],
       [3.8],
       [6.9],
       [7.8]])
```

In [11]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(X,y,random_state=1,test_size=0.2)
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
```

In [12]:
```
regressor.fit(xtrain,ytrain)
```

Out[12]:
```
LinearRegression()
```

In [13]:
```
print(regressor.coef_)
```

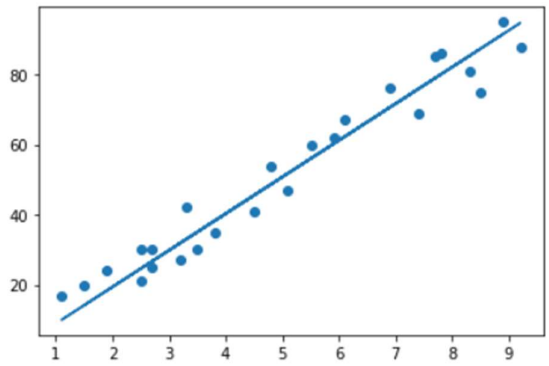
```
[10.46110829]
```

In [14]:
```
print(regressor.intercept_)
```

```
-1.5369573315500702
```

In [15]:
```
line=regressor.coef_*X+regressor.intercept_
plt.scatter(X,y)
plt.plot(X,line)
line
```

Out[15]:
```
array([[24.6158134 ],
       [51.81469497],
       [31.93858921],
       [87.38246316],
       [35.0769217 ],
       [14.15470511],
       [94.70523897],
       [55.99913828],
       [85.29024151],
       [26.70803506],
       [79.01357653],
       [60.1835816 ],
       [45.53802999],
       [32.98470004],
       [ 9.97026179],
       [91.56690648],
       [24.6158134 ],
       [18.33914843],
       [62.27580326],
       [75.87524404],
       [26.70803506],
       [48.67636248],
       [38.21525418],
       [70.64468989],
       [80.05968736]])
```



In [17]:
```
ypred=regressor.predict(xtest)
ypred
```

Out[17]:
```
array([ 9.97026179, 32.98470004, 18.33914843, 87.38246316, 48.67636248])
```

In [19]:
```
df=pd.DataFrame({'Actual':ytest,'Predicted':ypred})
df
```

Out[19]:

	Actual	Predicted
0	17	9.970262
1	42	32.984700
2	24	18.339148
3	75	87.382463
4	54	48.676362

In [20]:
```
from sklearn.metrics import mean_absolute_error
print(mean_absolute_error(ytest,ypred))
```

```
7.882398086270432
```