In [31]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [32]:

```
data=pd.read_csv('/content/student_scores.csv')
```

In [33]:

```
data.head()
```

Out[33]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

In [34]:

```
data.tail()
```

Out[34]:

	Hours	Scores
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [35]:

```
x=data.iloc[:,:-1].values
y=data.iloc[:,1].values
```

In [36]:

```
print(x)
```

```
[[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 [37]:

```
print(y)
```

```
[21 47 27 75 30 20 88 60 81 25 85 62 41 42 17 95 3
0 24 67 69 30 54 35 76
 86]
```

In [38]:

```
from sklearn.linear_model import LinearRegressio
from sklearn.model_selection import train_test_s
```

In [39]:

```
x_train,x_test,y_train,y_test=train_test_split(x
```

In [40]:

```
regressor=LinearRegression()
regressor.fit(x_train,y_train)
```

Out[40]:

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [41]:

```
y_pred=regressor.predict(x_test)
print(y_pred)
```

```
[17.04289179 33.51695377 74.21757747 26.73351648 5
9.68164043 39.33132858
 20.91914167 78.09382734 69.37226512]
```

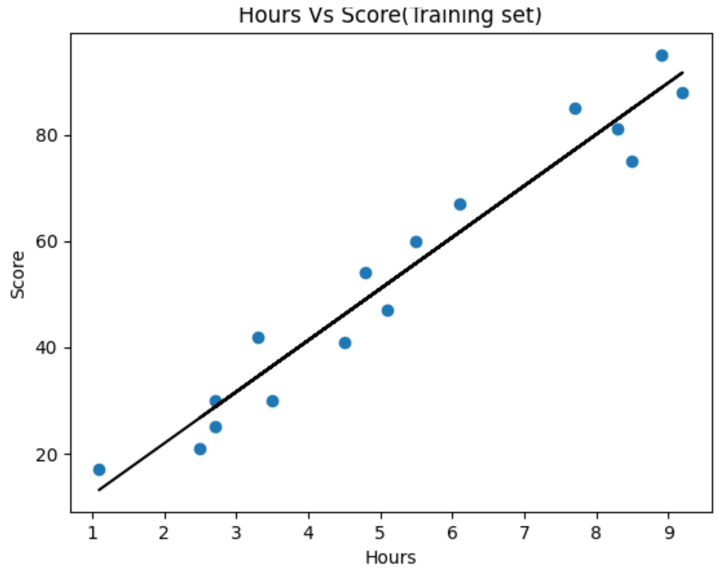
In [42]:

```
print(y_test)
```

```
[20 27 69 30 62 35 24 86 76]
```

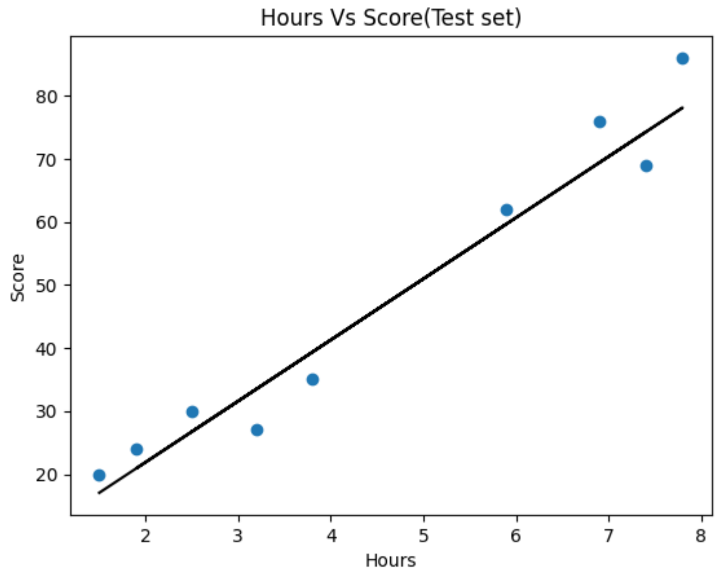
In [43]:

```
#for train values
plt.scatter(x_train,y_train)
plt.plot(x_train,regressor.predict(x_train),colo
plt.title("Hours Vs Score(Training set)")
plt.xlabel("Hours")
plt.ylabel("Score")
plt.show()
```



In [44]:

```
#for test values
y_pred=regressor.predict(x_test)
plt.scatter(x_test,y_test)
plt.plot(x_test,regressor.predict(x_test),color=
plt.title("Hours Vs Score(Test set)")
plt.xlabel("Hours")
plt.ylabel("Score")
plt.show()
```



In [45]:

```
import sklearn.metrics as metrics

mae = metrics.mean_absolute_error(x, y)
mse = metrics.mean_squared_error(x, y)
rmse = np.sqrt(mse)

print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
```

```
MAE: 46.468
MSE: 2659.5692
```

```
MSE: 2659.5692
RMSE: 51.57101123693426
```