Task 1

In [1]:
```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
```

importing data

In [2]:
```
dataset = pd.read_csv('https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores%20-%20student_scores.csv')
```

In [3]:
```
dataset
```

Out[3]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [4]:
```
dataset.head()
```

Out[4]:

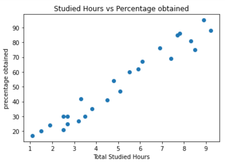
	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

In [5]:
```
dataset.shape
```

Out[5]: (25, 2)

Visualization

In [6]:
```
plt.scatter(dataset['Hours'], dataset['Scores'])
plt.title('Studied Hours vs Percentage obtained')
plt.xlabel('Total Studied Hours')
plt.ylabel('precentage obtained')
plt.show()
```



spliting data for trainig model

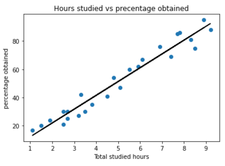
In [18]:
```
X = dataset.iloc[:,:-1].values
Y = dataset.iloc[:,-1].values
```

In [19]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3, random_state = 0)
regression = LinearRegression()
regression.fit(X_train, Y_train)
```

Out[19]: LinearRegression()

Plotting regression line

In [21]:
```
line = regression.coef_*X + regression.intercept_
plt.scatter(X, Y)
plt.plot(X, line, color = 'black')
plt.title('Hours studied vs precentage obtained')
plt.xlabel('Total studied hours')
plt.ylabel('percentage obtained')
plt.show()
```

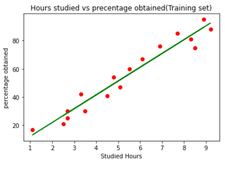


In [22]:
```
Y_pred = regression.predict(X_test)
print(Y_pred)
```

```
[17.05366541 33.69422878 74.80620886 26.8422321  60.12335883 39.56736879
 20.96909209 78.72163554]
```

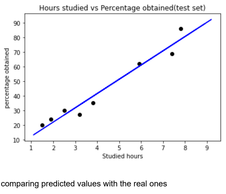
Visualizing training set

In [23]:
```
plt.scatter(X_train, Y_train, color = 'red')
plt.plot(X_train, regression.predict(X_train), color = 'green')
plt.title('Hours studied vs precentage obtained(Training set)')
plt.xlabel('Studied Hours')
plt.ylabel('percentage obtained')
plt.show()
```



Visualization for test set

In [25]:
```
plt.scatter(X_test, Y_test, color = 'black')
plt.plot(X_train, regression.predict(X_train),color = 'blue')
plt.title('Hours studied vs Percentage obtained(test set)')
plt.xlabel('Studied hours')
plt.ylabel('percentage obtained')
plt.show()
```



comparing predicted values with the real ones

In [26]:
```
dataset = pd.DataFrame({'Actual values': Y_test, 'predicted value': Y_pred})
dataset
```

Out[26]:

	Actual values	predicted value
0	20	17.053665
1	27	33.694229
2	69	74.806209
3	30	26.842232
4	62	60.123359
5	35	39.567369
6	24	20.969092
7	86	78.721636

Predicting score

In [33]:
```
dataset = np.array(9.25)
dataset = dataset.reshape(-1,1)
pred = regression.predict(dataset)
print('The score is()' .format(pred))
```

```
The score is[92.91505723]
```

Mean absolute error

In [36]:
```
from sklearn import metrics
print('Mean absolute error',metrics.mean_absolute_error(Y_test, Y_pred))
```

```
Mean absolute error 4.419727808027652
```

R square error

In [37]:
```
from sklearn.metrics import r2_score
print('R square error',r2_score(Y_test, Y_pred))
```

```
R square error 0.9568211104435257
```