In [16]:
```
# Import necessary libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
```

In [17]:
```
# Loading the dataset
df = pd.read_csv('C:\\Users\\hm\\Desktop\\jat\\student_scores_dataset.csv')
```

In [18]:
```
df
```

Out[18]:

	Study Hours	Exam Scores
0	3.7	87.9
1	9.5	143.6
2	7.3	123.7
3	6.0	99.9
4	1.6	64.5
...	...	...
95	4.9	95.3
96	5.2	101.9
97	4.3	94.5
98	0.3	53.9
99	1.1	64.9

100 rows × 2 columns

In [42]:
```
x = np.array(df["Study Hours"]).reshape(-1,1)
y = np.array(df["Exam Scores"])
```

In [43]:
```
x
```

Out[43]:
```
array([[3.7],
       [9.5],
       [7.3],
       [6. ],
       [1.6],
       [1.6],
       [0.6],
       [8.7],
       [6. ],
       [7.1],
       [0.2],
       [9.7],
       [8.3],
       [2.1],
       [1.8],
       [1.8],
       [3. ],
       [5.2],
       [4.3],
       [2.9],
       [6.1],
       [1.4],
       [2.9],
       [3.7],
       [4.6],
       [7.9],
       [2. ],
       [5.1],
       [5.9],
       [0.5],
       [6.1],
       [1.7],
       [0.7],
       [9.5],
       [9.7],
       [8.1],
       [3. ],
       [1. ],
       [6.8],
       [4.4],
       [1.2],
       [5. ],
       [0.3],
       [9.1],
       [2.6],
       [6.6],
       [3.1],
       [5.2],
       [5.5],
       [1.8],
       [9.7],
       [7.8],
       [9.4],
       [8.9],
       [6. ],
       [9.2],
       [0.9],
```

```
       [2. ],
       [0.5],
       [3.3],
       [3.9],
       [2.7],
       [8.3],
       [3.6],
       [2.8],
       [5.4],
       [1.4],
       [8. ],
       [0.7],
       [9.9],
       [7.7],
       [2. ],
       [0.1],
       [8.2],
       [7.1],
       [7.3],
       [7.7],
       [0.7],
       [3.6],
       [1.2],
       [8.6],
       [6.2],
       [3.3],
       [0.6],
       [3.1],
       [3.3],
       [7.3],
       [6.4],
       [8.9],
       [4.7],
       [1.2],
       [7.1],
       [7.6],
       [5.6],
       [7.7],
       [4.9],
       [5.2],
       [4.3],
       [0.3],
       [1.1]])
```

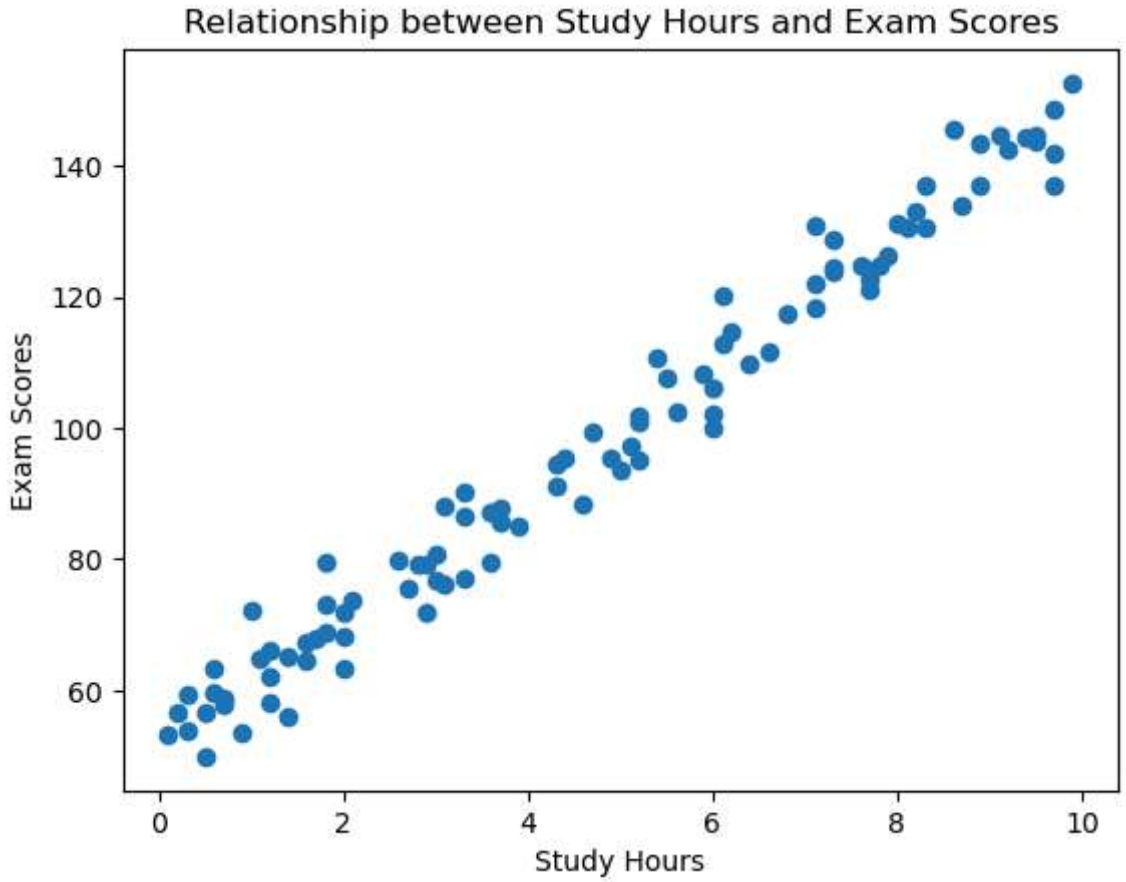
In [44]:
```
y
```

Out[44]:
```
array([ 87.9, 143.6, 123.7,  99.9,  64.5,  67.4,  63.2, 134. , 106.1,
       118.3,  56.6, 148.6, 130.6,  73.8,  68.7,  73.2,  76.9, 100.8,
        91.2,  71.8, 112.7,  65.3,  79.2,  85.5,  88.5, 126.4,  68.3,
        97.4, 108.4,  56.7, 120.2,  67.9,  57.8, 144.5, 137. , 130.7,
        80.8,  72.1, 117.5,  95.5,  62. ,  93.7,  59.2, 144.7,  79.8,
       111.7,  88.2,  95. , 107.6,  79.4, 142. , 124.7, 144.4, 137. ,
       102. , 142.5,  53.5,  72. ,  49.9,  90.3,  85. ,  75.5, 136.9,
        79.5,  79.2, 110.8,  56.1, 131.1,  58.8, 152.6, 121. ,  63.3,
        53.2, 133. , 121.9, 124.6, 123.7,  58.6,  87.3,  58. , 145.6,
       114.7,  77.1,  59.6,  76.2,  86.5, 128.8, 109.7, 143.5,  99.3,
        66.1, 130.8, 124.9, 102.4, 122.6,  95.3, 101.9,  94.5,  53.9,
        64.9])
```

In [22]:
```
#  Checking for missing values
print(df.isnull().sum())
```

```
Study Hours    0
Exam Scores    0
dtype: int64
```

In [25]:
```
# Visualizing the relationship between study hours and exam scores
plt.scatter(df['Study Hours'], df['Exam Scores'])
plt.title('Relationship between Study Hours and Exam Scores')
plt.xlabel('Study Hours')
plt.ylabel('Exam Scores')
plt.show()
```



In [26]:
```
#Spliting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rando
```

In [29]:
```
#Standardizing the independent variables
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [30]:
```
#Training a linear regression model
model = LinearRegression()
model.fit(X_train_scaled, y_train)
```

Out[30]:
```
LinearRegression
LinearRegression()
```

In [31]:
```
#Evaluating the performance of the model
y_pred = model.predict(X_test_scaled)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Absolute Error:", mae)
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

```
Mean Absolute Error: 2.9365732667749755
Mean Squared Error: 16.202109700645348
R-squared: 0.9826924926918468
```

In [33]:
```
#Interpreting the coefficients
coefficients = pd.DataFrame(model.coef_, index=['Study Hours'], columns=['Coef
print(coefficients)
```

```
             Coefficient
Study Hours    28.525561
```

In [39]:
```
#model coefficient
model.coef_
```

Out[39]:
```
array([28.52556103])
```

In [40]:
```
#model intercept

model.intercept_
```

Out[40]:
```
96.5875
```

In [ ]:
```

```