In [1]:

```
# Dataset Exploration Program:

from sklearn.datasets import load_iris
import pandas as pd

# Load the dataset
data = load_iris()
df = pd.DataFrame(data.data, columns=data.feature_names)

# Display the first five rows
print("First five rows:\n", df.head())

# Display the dataset's shape
print("\nDataset shape:", df.shape)

# Display summary statistics
print("\nSummary statistics:\n", df.describe())
```

```
First five rows:
    sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

Dataset shape: (150, 4)

Summary statistics:
        sepal length (cm)  sepal width (cm)  petal length (cm)  \
count         150.000000        150.000000         150.000000
mean            5.843333          3.057333           3.758000
std             0.828066          0.435866           1.765298
min             4.300000          2.000000           1.000000
25%             5.100000          2.800000           1.600000
50%             5.800000          3.000000           4.350000
75%             6.400000          3.300000           5.100000
max             7.900000          4.400000           6.900000

       petal width (cm)
count        150.000000
mean           1.199333
std            0.762238
min            0.100000
25%            0.300000
50%            1.300000
75%            1.800000
max            2.500000
```

In [2]:

```
# Data splitting Program:

from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris

# Load the dataset
data = load_iris()
X = data.data
y = data.target
```

```
# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ra

# Print the number of samples
print("Number of training samples:", X_train.shape[0])
print("Number of testing samples:", X_test.shape[0])
```

```
Number of training samples: 120
Number of testing samples: 30
```

In [3]:

```
# Linear Regression Program:

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np
import pandas as pd

# Sample dataset creation (for demonstration purposes)
# In practice, you would load your dataset here
data = pd.DataFrame({
    'YearsExperience': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'Salary': [40000, 45000, 50000, 55000, 60000, 65000, 70000, 75000, 8000
})

# Features and target variable
X = data[['YearsExperience']]
y = data['Salary']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ra

# Initialize and fit the model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict on the test set
y_pred = model.predict(X_test)

# Calculate Mean Squared Error
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error on the test set:", mse)
```

```
Mean Squared Error on the test set: 0.0
```