Experiment-11

11.Consider Mall_Customer Datset and perform the following operations

a)Using Elbow method find the optimal number of clusters

b) Train a K-Means cluster algorithm and Training dataset

c)Evaluate the model using inertia and other metrics (silhouette)

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
import numpy as np
```

1.Consider Mall_Customer Datset and perform the following operations

In [3]:

```
# Load dataset
df = pd.read_csv("C:/Users/Guru Kiran/All CSV files/Mall_Customers.csv")

print("Dataset Head:\n", df.head())
```

```
Dataset Head:
    CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
```

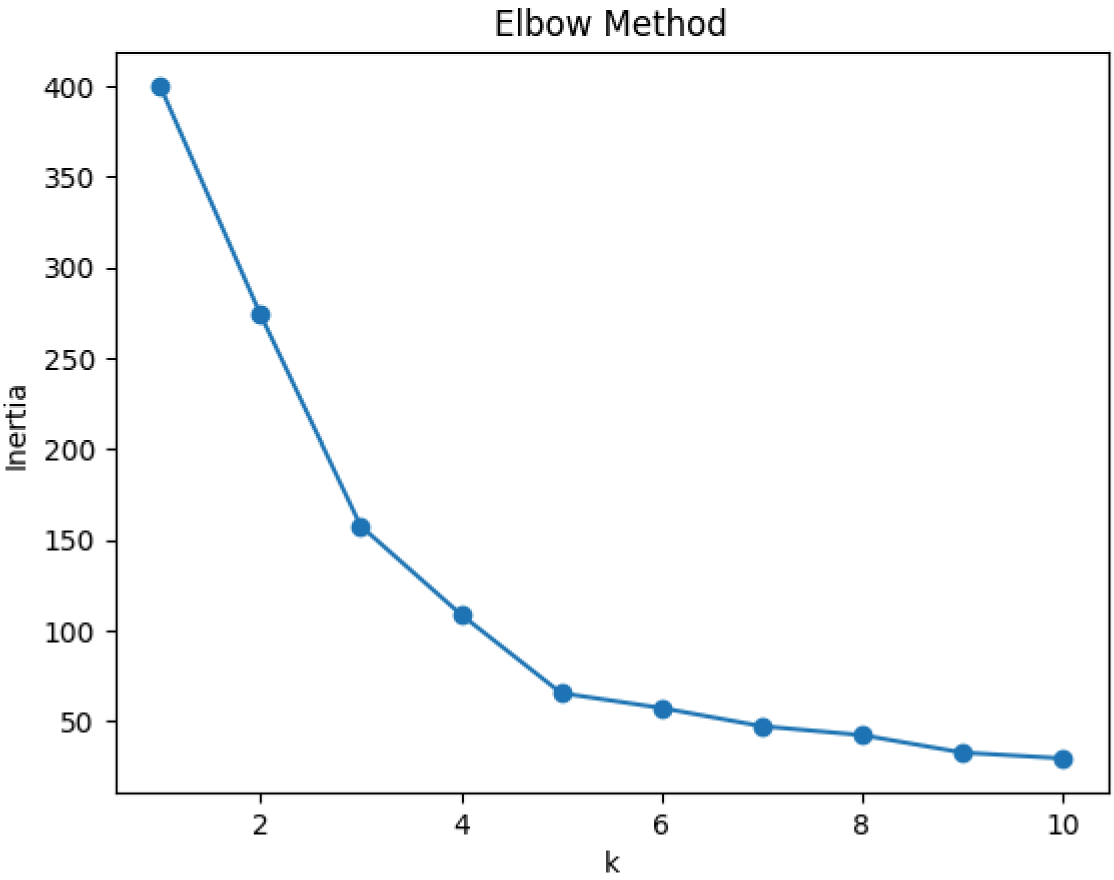
In [4]:

```
X = df[['Annual Income (k$)', 'Spending Score (1-100)']].values
X = StandardScaler().fit_transform(X)
```

a)Using Elbow method find the optimal number of clusters

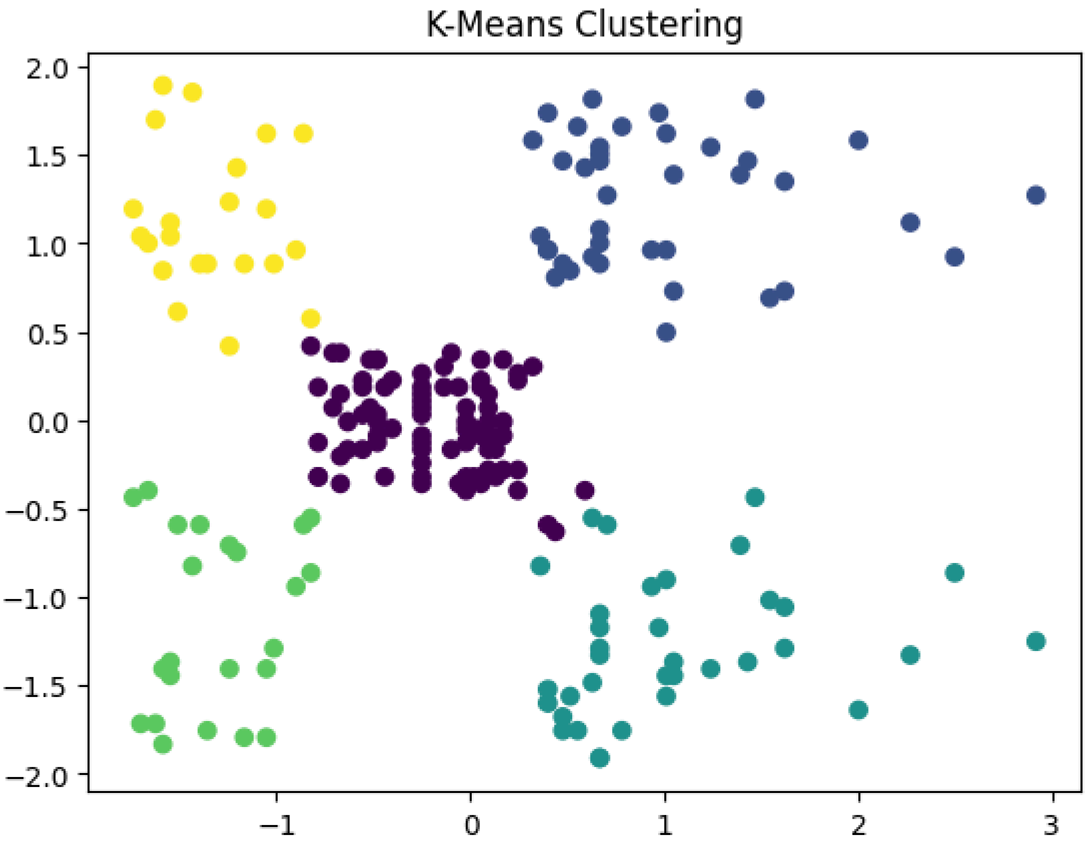
In [6]:

```
#inertias
inertias = []
for k in range(1, 11):
    km = KMeans(n_clusters=k, random_state=0)
    km.fit(X)
    inertias.append(km.inertia_)

plt.plot(range(1, 11), inertias, marker='o')
plt.title("Elbow Method")
plt.xlabel("k")
plt.ylabel("Inertia")
plt.show()
```



b) Train a K-Means cluster algorithm and Training dataset

In [7]:
```
# ---- 2. K-Means ----
kmeans = KMeans(n_clusters=5, random_state=0)
labels = kmeans.fit_predict(X)

plt.scatter(X[:, 0], X[:, 1], c=labels)
plt.title("K-Means Clustering")
plt.show()
```



c)Evaluate the model using inertia and other metrics (silhouette)

```
In [8]: # Silhouette Index
        score = silhouette_score(X, labels)
        print('Silhouette Index:', score)
```

```
Silhouette Index: 0.5546571631111091
```