Making k -means clustering project

In [19]:

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [20]:

```
# To Importing the dataset
dataset = pd.read_csv(r"C:\Users\HP\Downloads\Machine Learning\7 July, Hierarchical_clustering,K_means_cluster
x = dataset.iloc[:,[3,4]].values
```

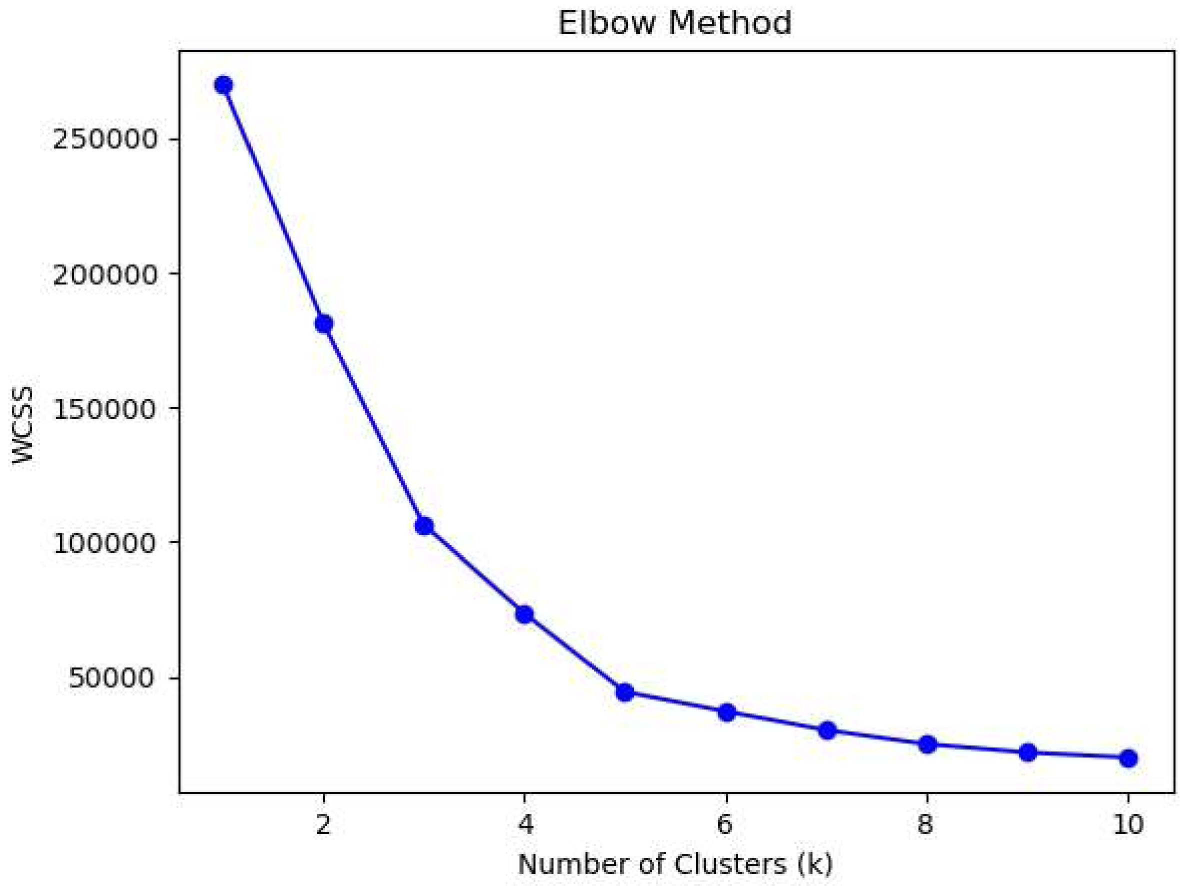
Use elbow method for finding the optimal number of clusters

In [24]:

```
import warnings
# Ignore all Warnings:
warnings.filterwarnings("ignore")
from sklearn.cluster import KMeans
# Assuming that you have stored data in 'x'
# x should be a 2D array or matrix with shape (n_samples, n_features)
# Initialize an empty list to store the WCSS values for different numbers of clusters
wcss = []
# Define the range of cluster numbers from for try
k_values = range (1,11) # Try cluster numbers from 1 to 10

# Calculate WCSS for each cluster number
for k in k_values:
    kmeans = KMeans(n_clusters = k, random_state = 42)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_) # Inertia is the WCSS value

# To Plot the WCSS values against the number of clusters:
plt.title('Elbow Method')
plt.plot(k_values,wcss, 'bo-')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('WCSS')
plt.show()
```

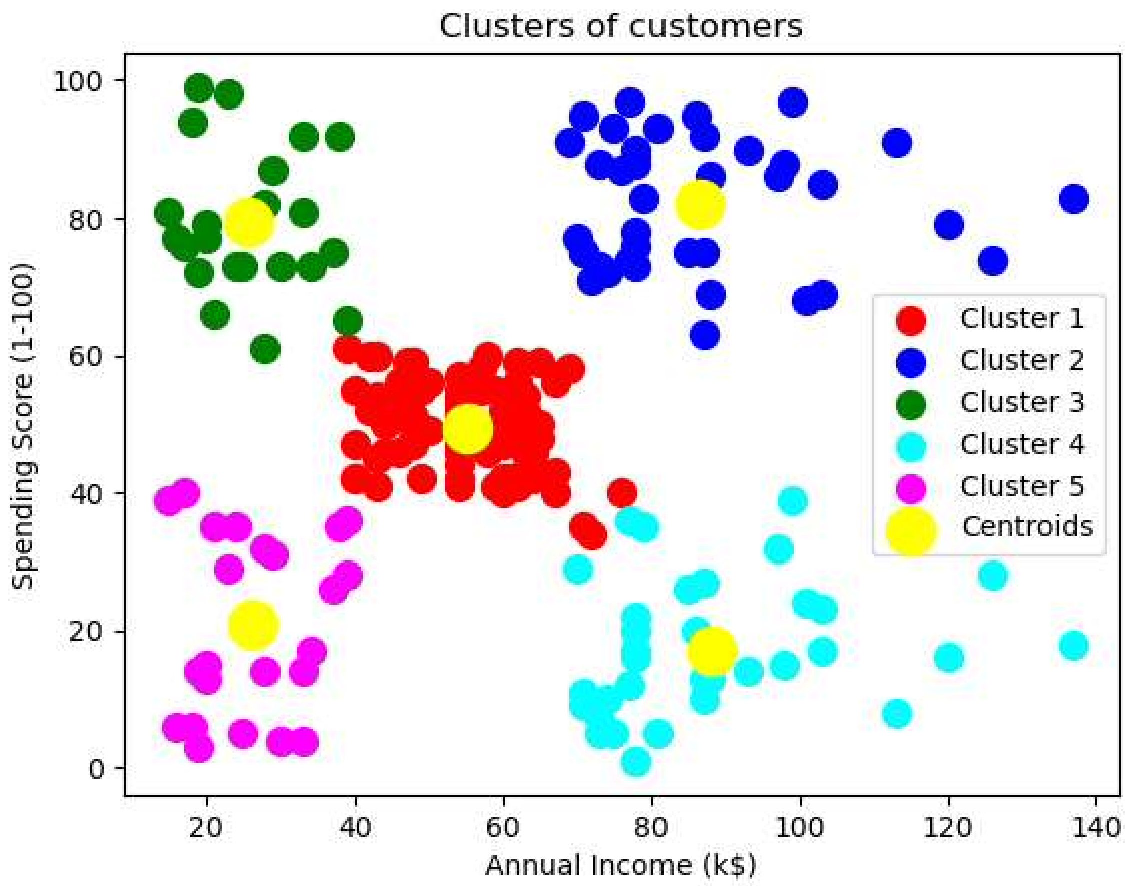


Training the k-means model on the dataset

In [25]:
```
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(x)
y_kmeans
```

Out[25]:
```
array([4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2,
       4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 0,
       4, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 1, 0, 1, 3, 1, 3, 1,
       0, 1, 3, 1, 3, 1, 3, 1, 3, 1, 0, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1,
       3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1,
       3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1,
       3, 1])
```

In [29]:
```
izing the clusters
ter(x[y_kmeans == 0, 0], x[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
ter(x[y_kmeans == 1, 0], x[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
ter(x[y_kmeans == 2, 0], x[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
ter(x[y_kmeans == 3, 0], x[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
ter(x[y_kmeans == 4, 0], x[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
ter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroids')
e('Clusters of customers')
el('Annual Income (k$)')
el('Spending Score (1-100)')
nd()
()
```



In [32]:
```
dataset['Cluster'] = kmeans.labels_
dataset.to_csv('modified_dataset.csv', index = False)
# Replace ('modified_dataset.csv' with the desire filename)
```

In [34]:
```
dataset.to_csv('modified_dataset.csv', index = False)
```

In [35]:
```
dataset.head()
```

Out[35]:

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

In [33]:

```
dataset['kmeans'] = y_kmeans
dataset.head()
```

Out[33]:

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

In [ ]: