# Name - Sarvesh Karanjkar
# PRN - 20210812002
# BDA LAB 05 (K-Means)

In [ ]:

```
# import required libraries#######
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt
%matplotlib inline
```

In [2]:

```
######### CREATING DATA FRAME

df = pd.read_csv("income.csv")
df.head()

###### WE ARE USING INCOME DATASET
```

Out[2]:

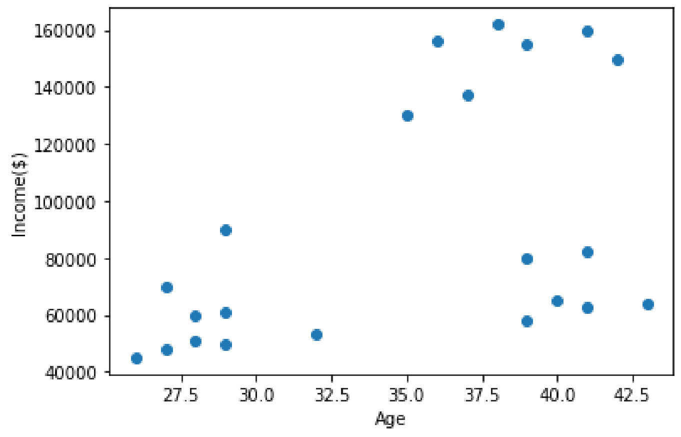
	Name	Age	Income($)
0	Rob	27	70000
1	Michael	29	90000
2	Mohan	29	61000
3	Ismail	28	60000
4	Kory	42	150000

In [3]:

```
######## PLOTTING SCATTER PLOT
plt.scatter(df.Age,df['Income($)'])
plt.xlabel('Age')
plt.ylabel('Income($)')
```

Out[3]:

```
Text(0, 0.5, 'Income($)')
```



In [4]:

```
######## APPLYING K-MEANS ON GIVEN AGE, INCOOME ATTRIBUTE.....
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age','Income($)']])
y_predicted
```

```
C:\Users\sarvesh\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: Th
e default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` ex
plicitly to suppress the warning
  warnings.warn(
C:\Users\sarvesh\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMe
ans is known to have a memory leak on Windows with MKL, when there are less chunks than availa
ble threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[4]:

```
array([2, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1])
```

In [6]:

```
df['cluster']=y_predicted
df.head()
```

Out[6]:

	Name	Age	Income($)	cluster
0	Rob	27	70000	2
1	Michael	29	90000	2
2	Mohan	29	61000	1
3	Ismail	28	60000	1
4	Kory	42	150000	0

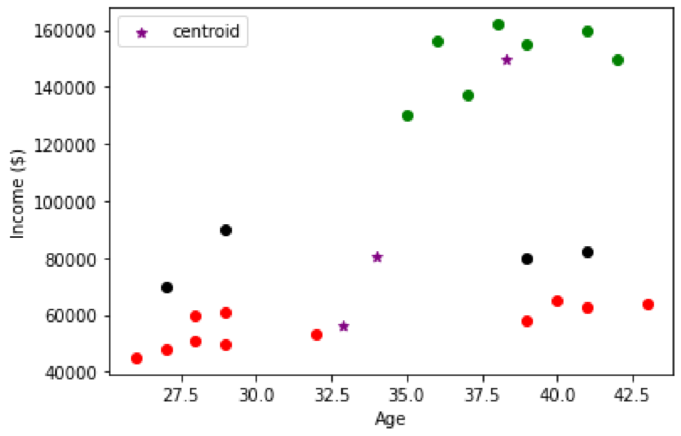
In [7]:

```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='black')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='centroid')
plt.xlabel('Age')
plt.ylabel('Income ($)')
plt.legend()

##### OBSERVED THAT CLUSTER IS IMPROPER
#### HERE WE NEED DAATA PRE PROCESSINGGG...
```

Out[7]:

```
<matplotlib.legend.Legend at 0x1eaac8d9a30>
```



In [9]:

```
######### HERE WE USED MIN-MAX SCLER IT TRANSFORM VALUES BETWEEN 0-1
scaler = MinMaxScaler()

scaler.fit(df[['Income($)']])
df['Income($)'] = scaler.transform(df[['Income($)']])

scaler.fit(df[['Age']])
df['Age'] = scaler.transform(df[['Age']])

df.head()
```

Out[9]:

	Name	Age	Income($)	cluster
0	Rob	0.058824	0.213675	2
1	Michael	0.176471	0.384615	2
2	Mohan	0.176471	0.136752	1
3	Ismail	0.117647	0.128205	1
4	Kory	0.941176	0.897436	0

In [10]:

```
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age','Income($)']])
y_predicted
```

```
C:\Users\sarvesh\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: Th
e default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` ex
plicitly to suppress the warning
  warnings.warn(
C:\Users\sarvesh\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMe
ans is known to have a memory leak on Windows with MKL, when there are less chunks than availa
ble threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[10]:

```
array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2])
```

In [11]:

```
### STORING VALUES TO DATAFRAME
df['cluster']=y_predicted
df.head()
```

Out[11]:

	Name	Age	Income($)	cluster
0	Rob	0.058824	0.213675	0
1	Michael	0.176471	0.384615	0
2	Mohan	0.176471	0.136752	0
3	Ismail	0.117647	0.128205	0
4	Kory	0.941176	0.897436	1

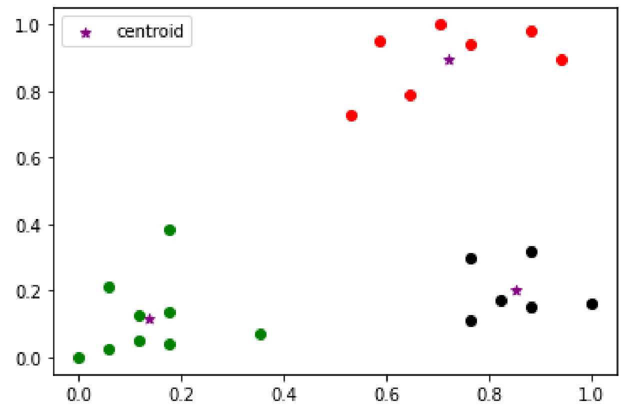
In [12]:

```
##### PLOTTING FINAL CLUSTERSSS ON GRAPH..

df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='black')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='centroid')
plt.legend()
```

Out[12]:

```
<matplotlib.legend.Legend at 0x1eaac957520>
```



## conclusion - performed K-means clustering using python on income data and created 3 clusters according to age and income with the help of kmeans library in python..