In [1]:

```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

In [2]:

```
df=pd.read_csv(r"C:\Users\shaik\Downloads\Income.csv")
df
```

Out[2]:

	Gender	Age	Income($)
0	Male	19	15
1	Male	21	15
2	Female	20	16
3	Female	23	16
4	Female	31	17
...	...	...	...
195	Female	35	120
196	Female	45	126
197	Male	32	126
198	Male	32	137
199	Male	30	137

200 rows × 3 columns

In [3]:

```
df.head
```

Out[3]:

```
<bound method NDFrame.head of      Gender  Age  Income($)
0      Male   19         15
1      Male   21         15
2    Female   20         16
3    Female   23         16
4    Female   31         17
..      ...  ...        ...
195  Female   35        120
196  Female   45        126
197    Male   32        126
198    Male   32        137
199    Male   30        137

[200 rows x 3 columns]>
```

In [4]:

```
df.tail
```

Out[4]:

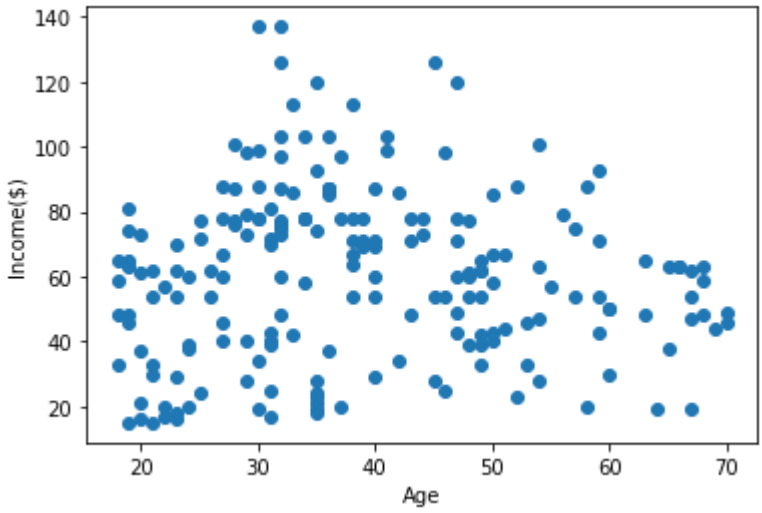
```
<bound method NDFrame.tail of      Gender  Age  Income($)
0      Male   19         15
1      Male   21         15
2    Female   20         16
3    Female   23         16
4    Female   31         17
..      ...  ...        ...
195  Female   35        120
196  Female   45        126
197    Male   32        126
198    Male   32        137
199    Male   30        137

[200 rows x 3 columns]>
```

In [5]:

```
plt.scatter(df["Age"],df["Income($)"])
plt.xlabel("Age")
plt.ylabel("Income($)")
```

Out[5]:

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



In [6]:

```
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[6]:

```
KMeans()
```

In [7]:

```
y_predicted=km.fit_predict(df[["Age","Income($)"]])
y_predicted
```

Out[7]:

```
array([5, 5, 5, 5, 5, 5, 5, 5, 1, 5, 1, 5, 1, 5, 5, 5, 5, 5, 1, 5, 5, 5,
       1, 5, 1, 5, 1, 5, 1, 5, 1, 5, 1, 5, 1, 5, 1, 5, 1, 5, 1, 5, 1, 4,
       1, 4, 1, 4, 4, 4, 1, 4, 4, 1, 1, 1, 1, 3, 4, 1, 3, 4, 3, 3, 3, 4,
       6, 3, 4, 4, 3, 6, 3, 3, 3, 4, 6, 6, 4, 6, 3, 6, 3, 6, 4, 6, 6, 4,
       4, 6, 3, 4, 6, 6, 4, 4, 6, 4, 6, 4, 4, 6, 3, 4, 6, 4, 3, 6, 3, 3,
       3, 4, 6, 4, 4, 4, 3, 6, 6, 6, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 6, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 7, 7, 7, 7, 7, 7,
       7, 7])
```

In [8]:

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

Out[8]:

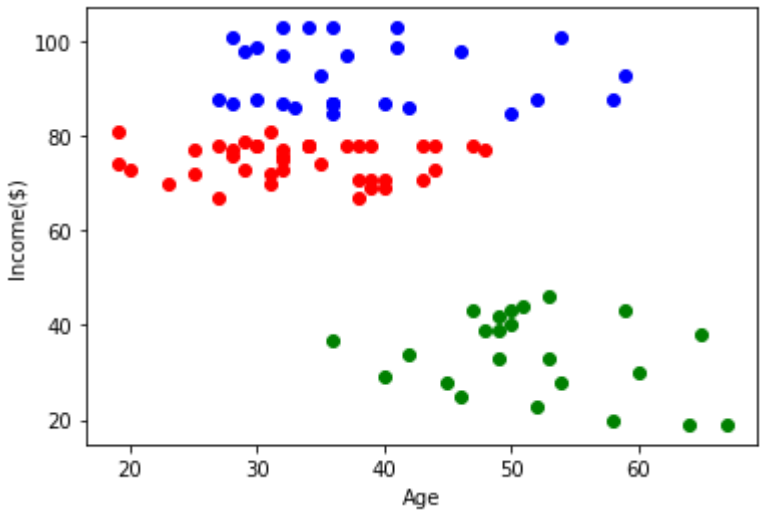
	Gender	Age	Income($)	cluster
0	Male	19	15	5
1	Male	21	15	5
2	Female	20	16	5
3	Female	23	16	5
4	Female	31	17	5

In [9]:

```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["Age"],df1["Income($)"],color="red")
plt.scatter(df2["Age"],df2["Income($)"],color="green")
plt.scatter(df3["Age"],df3["Income($)"],color="blue")
plt.xlabel("Age")
plt.ylabel("Income($)")
```

Out[9]:

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



In [10]:

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(df[["Income($)"]])
df["Income($)"]=scaler.transform(df[["Income($)"]])
df.head()
```

Out[10]:

	Gender	Age	Income($)	cluster
0	Male	19	0.000000	5
1	Male	21	0.000000	5
2	Female	20	0.008197	5
3	Female	23	0.008197	5
4	Female	31	0.016393	5

In [11]:

```
scaler.fit(df[["Age"]])
df["Age"]=scaler.transform(df[["Age"]])
df.head()
```

Out[11]:

	Gender	Age	Income($)	cluster
0	Male	0.019231	0.000000	5
1	Male	0.057692	0.000000	5
2	Female	0.038462	0.008197	5
3	Female	0.096154	0.008197	5
4	Female	0.250000	0.016393	5

In [13]:

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

Out[13]:

```
array([7, 7, 7, 7, 4, 7, 4, 7, 0, 4, 0, 4, 3, 7, 4, 7, 4, 7, 3, 4, 4, 7,
       3, 4, 3, 4, 3, 4, 4, 7, 0, 7, 3, 7, 3, 7, 3, 4, 4, 7, 0, 7, 3, 4,
       3, 7, 3, 4, 4, 4, 3, 4, 4, 0, 3, 3, 3, 0, 1, 3, 0, 1, 0, 3, 0, 1,
       3, 0, 1, 4, 0, 3, 0, 0, 0, 1, 3, 3, 1, 3, 0, 2, 0, 3, 1, 3, 6, 1,
       2, 6, 0, 1, 6, 2, 2, 1, 6, 1, 6, 1, 1, 6, 0, 1, 6, 1, 0, 6, 0, 0,
       0, 1, 2, 1, 1, 1, 0, 6, 6, 6, 1, 2, 2, 2, 1, 2, 6, 2, 6, 2, 6, 2,
       1, 2, 1, 2, 6, 2, 1, 2, 6, 2, 2, 2, 1, 2, 6, 2, 2, 2, 6, 2, 6, 2,
       6, 2, 2, 2, 2, 2, 6, 2, 1, 2, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 2,
       6, 2, 6, 2, 5, 5, 6, 5, 5, 5, 6, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5])
```

In [14]:

```
df["New Cluster"]=y_predicted
df.head()
```

Out[14]:

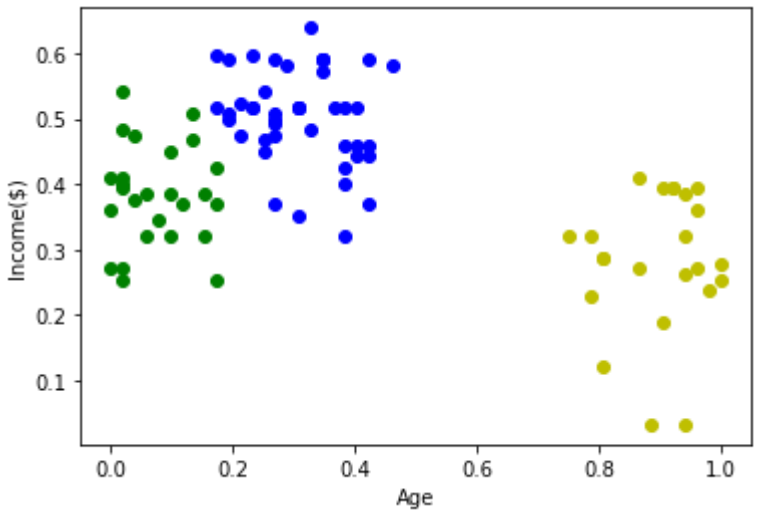
	Gender	Age	Income($)	cluster	New Cluster
0	Male	0.019231	0.000000	5	7
1	Male	0.057692	0.000000	5	7
2	Female	0.038462	0.008197	5	7
3	Female	0.096154	0.008197	5	7
4	Female	0.250000	0.016393	5	4

In [16]:

```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["Age"],df1["Income($)"],color="y")
plt.scatter(df2["Age"],df2["Income($)"],color="green")
plt.scatter(df3["Age"],df3["Income($)"],color="blue")
plt.xlabel("Age")
plt.ylabel("Income($)")
```

Out[16]:

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



In [17]:

```
km.cluster_centers_
```

Out[17]:

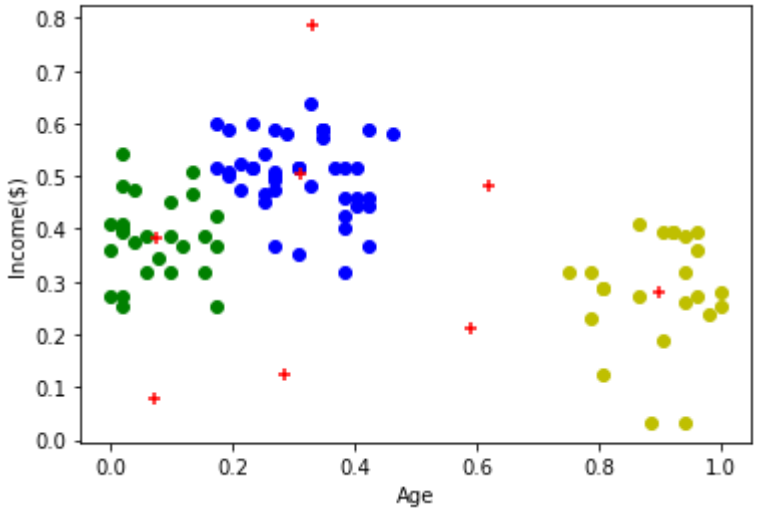
```
array([[0.89799331, 0.28011404],
       [0.07322485, 0.38272383],
       [0.30944056, 0.50428465],
       [0.58974359, 0.20969945],
       [0.28388278, 0.1245121 ],
       [0.32905983, 0.78551913],
       [0.62037037, 0.47996357],
       [0.07239819, 0.08003857]])
```

In [19]:

```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["Age"],df1["Income($)"],color="y")
plt.scatter(df2["Age"],df2["Income($)"],color="green")
plt.scatter(df3["Age"],df3["Income($)"],color="blue")
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color="red",marker="+")
plt.xlabel("Age")
plt.ylabel("Income($)")
```

Out[19]:

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



In [20]:

```
k_rng=range(1,10)
sse=[]
```

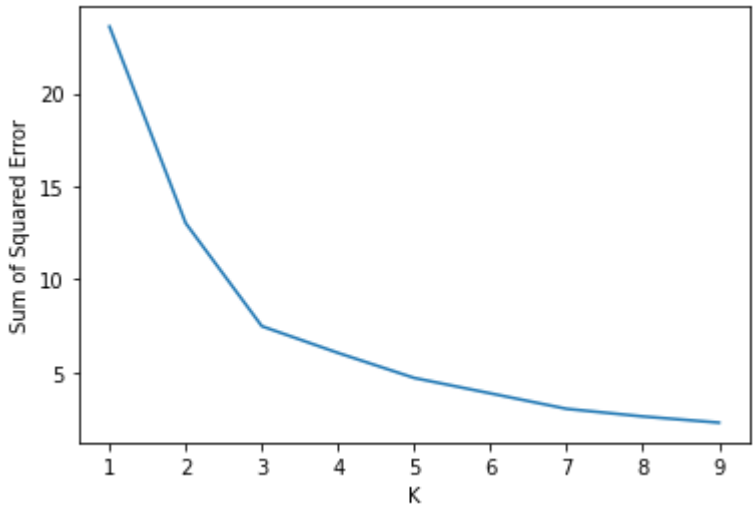
In [21]:

```
for k in k_rng:
 km=KMeans(n_clusters=k)
 km.fit(df[["Age","Income($)"]])
 sse.append(km.inertia_)
#km.inertia_ will give you the value of sum of square error
print(sse)
plt.plot(k_rng,sse)
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")
```

```
C:\Users\shaik\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:103
6: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(

[23.583906150363603, 13.028938428018286, 7.492113413237458, 6.058372453353
154, 4.71335402850073, 3.8862179388020657, 3.054717436369358, 2.6460609774
305146, 2.3135720353543285]
```

Out[21]:

```
Text(0, 0.5, 'Sum of Squared Error')
```



In [ ]: