In [1]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import LabelEncoder
```

In [2]:

```
data=pd.read_csv('apples_and_oranges.csv')
print(data)
```

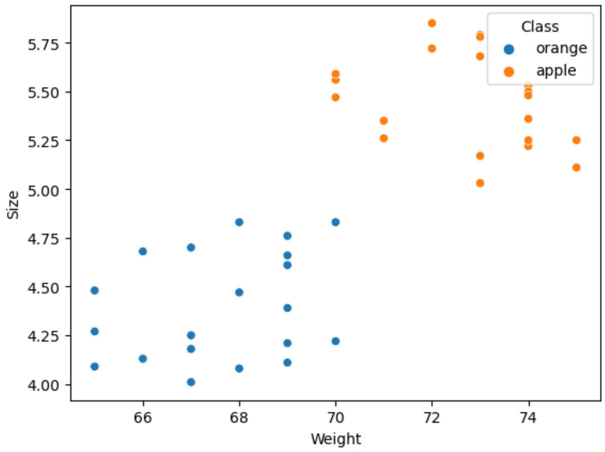
```
    Weight  Size   Class
0       69  4.39  orange
1       69  4.21  orange
2       65  4.09  orange
3       72  5.85   apple
4       67  4.70  orange
5       73  5.68   apple
6       70  5.56   apple
7       75  5.11   apple
8       74  5.36   apple
9       65  4.27  orange
10      73  5.79   apple
11      70  5.47   apple
12      74  5.53   apple
13      68  4.47  orange
14      74  5.22   apple
15      65  4.48  orange
16      69  4.66  orange
17      75  5.25   apple
18      67  4.18  orange
19      74  5.50   apple
20      66  4.13  orange
21      70  4.83  orange
22      69  4.61  orange
23      68  4.08  orange
24      67  4.25  orange
25      71  5.35   apple
26      67  4.01  orange
27      70  4.22  orange
28      74  5.25   apple
29      71  5.26   apple
30      73  5.78   apple
31      66  4.68  orange
32      72  5.72   apple
33      73  5.17   apple
34      68  4.83  orange
35      69  4.11  orange
36      69  4.76  orange
37      74  5.48   apple
38      70  5.59   apple
39      73  5.03   apple
```

In [3]:

```
import seaborn as sns
sns.scatterplot(x='Weight',y='Size',hue='Class',data=data)
```

Out[3]:

```
<AxesSubplot:xlabel='Weight', ylabel='Size'>
```



In [10]:
```
train_set,test_set=train_test_split(data,test_size=0.2,random_state=1)
print("train:",train_set)
print("test:",test_set)
```

```
train:     Weight  Size   Class
19      74  5.50   apple
26      67  4.01  orange
32      72  5.72   apple
17      75  5.25   apple
30      73  5.78   apple
36      69  4.76  orange
33      73  5.17   apple
28      74  5.25   apple
4       67  4.70  orange
14      74  5.22   apple
10      73  5.79   apple
35      69  4.11  orange
23      68  4.08  orange
24      67  4.25  orange
34      68  4.83  orange
20      66  4.13  orange
18      67  4.18  orange
25      71  5.35   apple
6       70  5.56   apple
13      68  4.47  orange
7       75  5.11   apple
38      70  5.59   apple
1       69  4.21  orange
16      69  4.66  orange
0       69  4.39  orange
15      65  4.48  orange
5       73  5.68   apple
11      70  5.47   apple
9       65  4.27  orange
8       74  5.36   apple
12      74  5.53   apple
37      74  5.48   apple
test:     Weight  Size   Class
2       65  4.09  orange
31      66  4.68  orange
3       72  5.85   apple
21      70  4.83  orange
27      70  4.22  orange
29      71  5.26   apple
22      69  4.61  orange
39      73  5.03   apple
```

In [11]:
```
x_train=train_set.iloc[:,0:2].values
y_train=train_set.iloc[:,2].values
x_test=test_set.iloc[:,0:2].values
y_test=test_set.iloc[:,2].values
print(x_train,y_train)
print(x_test,y_test)
```

```
[[74.    5.5 ]
 [67.    4.01]
 [72.    5.72]
 [75.    5.25]
 [73.    5.78]
 [69.    4.76]
 [73.    5.17]
 [74.    5.25]
 [67.    4.7 ]
 [74.    5.22]
 [73.    5.79]
 [69.    4.11]
 [68.    4.08]
 [67.    4.25]
 [68.    4.83]
 [66.    4.13]
 [67.    4.18]
 [71.    5.35]
 [70.    5.56]
 [68.    4.47]
 [75.    5.11]
 [70.    5.59]
 [69.    4.21]
 [69.    4.66]
 [69.    4.39]
 [65.    4.48]
 [73.    5.68]
 [70.    5.47]
 [65.    4.27]
 [74.    5.36]
 [74.    5.53]
 [74.    5.48]] ['apple' 'orange' 'apple' 'apple' 'apple' 'orange' 'apple' 'apple'
 'orange' 'apple' 'apple' 'orange' 'orange' 'orange' 'orange' 'orange'
 'orange' 'apple' 'apple' 'orange' 'apple' 'apple' 'orange' 'orange'
 'orange' 'orange' 'apple' 'apple' 'orange' 'apple' 'apple' 'apple']
[[65.    4.09]
 [66.    4.68]
 [72.    5.85]
 [70.    4.83]
 [70.    4.22]
 [71.    5.26]
 [69.    4.61]
 [73.    5.03]] ['orange' 'orange' 'apple' 'orange' 'orange' 'apple' 'orange' 'apple']
```

In [6]:
```
c=SVC(kernel='rbf',random_state=1,C=1,gamma='auto')
c.fit(x_train,y_train)
```

Out[6]:
```
SVC(C=1, gamma='auto', random_state=1)
```

In [7]:
```
yp=c.predict(x_test)
print(yp)
```

```
['orange' 'orange' 'apple' 'apple' 'orange' 'apple' 'orange' 'apple']
```

In [8]:
```
cm=confusion_matrix(y_test,yp)
print(cm)
accuracy=float(cm.diagonal().sum())/len(y_test)
print("model accuracy is:",accuracy*100,'%')
```

```
[[3 0]
 [1 4]]
model accuracy is: 87.5 %
```

```
In [ ]:
```