Karanjot Singh

1. Perform Data preprocessing with the given data set (handling missing values etc.)

In [208]:
```
import pandas as pd
from sklearn import datasets, linear_model
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
import seaborn as sns
train_data = pd.read_csv('train.csv')
train_data['Age'].fillna(train_data['Age'].mean(),inplace=True)
print(train_data)
```

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                  Name     Sex        Age  \
0                              Braund, Mr. Owen Harris    male  22.000000
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.000000
2                               Heikkinen, Miss. Laina  female  26.000000
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.000000
4                             Allen, Mr. William Henry    male  35.000000
..                                                 ...     ...        ...
886                              Montvila, Rev. Juozas    male  27.000000
887                       Graham, Miss. Margaret Edith  female  19.000000
888           Johnston, Miss. Catherine Helen "Carrie"  female  29.699118
889                              Behr, Mr. Karl Howell    male  26.000000
890                                Dooley, Mr. Patrick    male  32.000000

     SibSp  Parch            Ticket     Fare Cabin Embarked
0        1      0         A/5 21171   7.2500   NaN        S
1        1      0          PC 17599  71.2833   C85        C
2        0      0  STON/O2. 3101282   7.9250   NaN        S
3        1      0            113803  53.1000  C123        S
4        0      0            373450   8.0500   NaN        S
..     ...    ...               ...      ...   ...      ...
886      0      0            211536  13.0000   NaN        S
887      0      0            112053  30.0000   B42        S
888      1      2        W./C. 6607  23.4500   NaN        S
889      0      0            111369  30.0000  C148        C
890      0      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

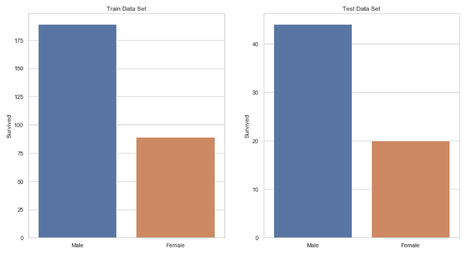
2. Perform Train Test split onto the data set using sklearn.

In [209]:
```
train_set,test_set = train_test_split(train_data, test_size = 0.2)
```

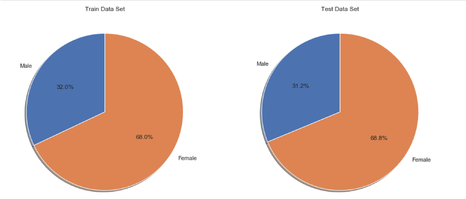
3. Show using plot and also calculate the survival as per gender (bar graph)and survival percentage as per gender(pie chart) (on both test & train data set).

In [210]:
```
survived_in_trainset = train_set[train_set['Survived'] == 1]
age = survived_in_trainset['Age']
male_survived_trainset  = len(survived_in_trainset[survived_in_trainset['Sex']=='male'])
female_survived_trainset = len(survived_in_trainset[survived_in_trainset['Sex']=='female'])
survived_train_grouped = survived_in_trainset.groupby('Sex').count()

survived_in_testset = test_set[test_set['Survived'] == 1]
male_survived_testset  = len(survived_in_testset[survived_in_testset['Sex']=='male'])
female_survived_testset = len(survived_in_testset[survived_in_testset['Sex']=='female'])
survived_test_grouped = survived_in_testset.groupby('Sex').count()

sns.set(style = 'whitegrid')
fig = plt.figure(figsize = (15,8))

ax2 = fig.add_subplot(1,2,1)
ax = sns.barplot(x = ['Male','Female'] , y = 'Survived',data = survived_train_grouped)

plt.title("Train Data Set")

ax2 = fig.add_subplot(1,2,2)
bx = sns.barplot(x = ['Male','Female'] , y = 'Survived',data = survived_test_grouped)
plt.title("Test Data Set")
```

Out[210]: Text(0.5, 1.0, 'Test Data Set')



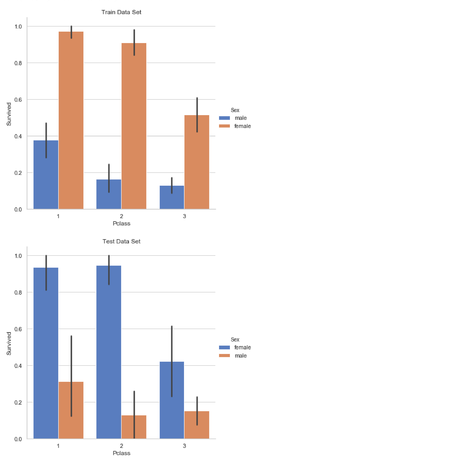
In [211]:
```
fig = plt.figure(figsize = (15,8))
ax2 = fig.add_subplot(1,2,1)
plt.title("Train Data Set")
ax2.pie([male_survived_trainset,female_survived_trainset], labels=['Male','Female'], autopct='%1.1f%%',shadow=True, startangle=9
0)
ax2 = fig.add_subplot(1,2,2)
plt.title("Test Data Set")
ax2.pie([male_survived_testset,female_survived_testset], labels=['Male','Female'], autopct='%1.1f%%',shadow=True, startangle=90)
plt.show()
```



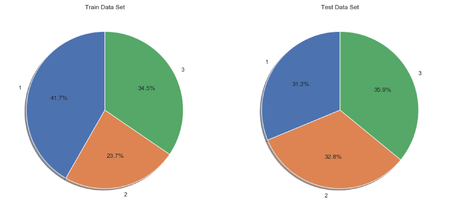
4. Show using plot and also calculate the survival as per Social Economic Status(SES) (bar graph)and survival percentage as per SES(pie chart) (on both test & train data set).

In [212]:
```
sns.set(style="whitegrid")
k = sns.catplot(x="Pclass", y="Survived", hue="Sex", data=train_set,
                height=6, kind="bar", palette="muted").set_titles('Train Data Set')
plt.title("Train Data Set")

sns.catplot(x="Pclass", y="Survived", hue="Sex", data=test_set,
                height=6, kind="bar", palette="muted")
plt.title("Test Data Set")
```

Out[212]: Text(0.5, 1, 'Test Data Set')



In [213]:
```
pclass_1_servived_train = survived_in_trainset[survived_in_trainset['Pclass'] == 1].count()
pclass_2_servived_train = survived_in_trainset[survived_in_trainset['Pclass'] == 2].count()
pclass_3_servived_train = survived_in_trainset[survived_in_trainset['Pclass'] == 3].count()
#print(pclass_1_servived.Pclass)
fig = plt.figure(figsize = (15,8))
ax2 = fig.add_subplot(1,2,1)
plt.title("Train Data Set")
ax2.pie([pclass_1_servived_train.Pclass,pclass_2_servived_train.Pclass,pclass_3_servived_train.Pclass],labels=['1','2','3'], aut
opct='%1.1f%%',shadow=True, startangle=90)


pclass_1_servived_test = survived_in_testset[survived_in_testset['Pclass'] == 1].count()
pclass_2_servived_test = survived_in_testset[survived_in_testset['Pclass'] == 2].count()
pclass_3_servived_test = survived_in_testset[survived_in_testset['Pclass'] == 3].count()

ax2 = fig.add_subplot(1,2,2)
plt.title("Test Data Set")
ax2.pie([pclass_1_servived_test.Pclass,pclass_2_servived_test.Pclass,pclass_3_servived_test.Pclass],labels=['1','2','3'], autopc
t='%1.1f%%',shadow=True, startangle=90)
plt.show()
```



5. Calculate the minimum survival age and maximum survival age on the train dataset

In [214]:
```
print("Max Age from Survival:",age.max())
#for i in range(len(age)):
#    if age[i] < 1:
#        age[i] = age[i]*100
print("Min Age from Survival:",age.min())
```

```
Max Age from Survival: 80.0
Min Age from Survival: 0.42
```