In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [6]:

```
df = pd.read_csv("Expanded_data_with_more_features.csv")
print(df.head())
```

```
   Unnamed: 0  Gender EthnicGroup          ParentEduc     LunchType TestPrep  \
0           0  female         NaN   bachelor's degree      standard     none   
1           1  female     group C        some college      standard      NaN   
2           2  female     group B     master's degree      standard     none   
3           3    male     group A  associate's degree  free/reduced     none   
4           4    male     group C        some college      standard     none   

  ParentMaritalStatus PracticeSport IsFirstChild  NrSiblings TransportMeans  \
0             married     regularly          yes         3.0     school_bus   
1             married     sometimes          yes         0.0            NaN   
2              single     sometimes          yes         4.0     school_bus   
3             married         never           no         1.0            NaN   
4             married     sometimes          yes         0.0     school_bus   

  WklyStudyHours  MathScore  ReadingScore  WritingScore  
0            < 5         71            71            74  
1         5 - 10         69            90            88  
2            < 5         87            93            91  
3         5 - 10         45            56            42  
4         5 - 10         76            78            75  
```

In [7]:

```
df.describe()
```

Out[7]:

	Unnamed: 0	NrSiblings	MathScore	ReadingScore	WritingScore
count	30641.000000	29069.000000	30641.000000	30641.000000	30641.000000
mean	499.556607	2.145894	66.558402	69.377533	68.418622
std	288.747894	1.458242	15.361616	14.758952	15.443525
min	0.000000	0.000000	0.000000	10.000000	4.000000
25%	249.000000	1.000000	56.000000	59.000000	58.000000
50%	500.000000	2.000000	67.000000	70.000000	69.000000
75%	750.000000	3.000000	78.000000	80.000000	79.000000
max	999.000000	7.000000	100.000000	100.000000	100.000000

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30641 entries, 0 to 30640
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype  
```

```
---  ------              --------------  -----
 0   Unnamed: 0          30641 non-null  int64
 1   Gender              30641 non-null  object
 2   EthnicGroup         28801 non-null  object
 3   ParentEduc          28796 non-null  object
 4   LunchType           30641 non-null  object
 5   TestPrep            28811 non-null  object
 6   ParentMaritalStatus  29451 non-null  object
 7   PracticeSport       30010 non-null  object
 8   IsFirstChild        29737 non-null  object
 9   NrSiblings          29069 non-null  float64
 10  TransportMeans      27507 non-null  object
 11  WklyStudyHours      29686 non-null  object
 12  MathScore           30641 non-null  int64
 13  ReadingScore        30641 non-null  int64
 14  WritingScore        30641 non-null  int64
dtypes: float64(1), int64(4), object(10)
memory usage: 3.5+ MB
```

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
Unnamed: 0                0
Gender                    0
EthnicGroup            1840
ParentEduc             1845
LunchType                 0
TestPrep               1830
ParentMaritalStatus    1190
PracticeSport           631
IsFirstChild            904
NrSiblings             1572
TransportMeans         3134
WklyStudyHours          955
MathScore                 0
ReadingScore              0
WritingScore              0
dtype: int64
```

drop unname column

In [12]:

```
df = df.drop("Unnamed: 0", axis = 1)
```

In [13]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30641 entries, 0 to 30640
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Gender              30641 non-null  object
 1   EthnicGroup         28801 non-null  object
 2   ParentEduc          28796 non-null  object
 3   LunchType           30641 non-null  object
 4   TestPrep            28811 non-null  object
 5   ParentMaritalStatus  29451 non-null  object
 6   PracticeSport        30010 non-null  object
 7   IsFirstChild         29737 non-null  object
 8   NrSiblings           29069 non-null  float64
 9   TransportMeans       27507 non-null  object
 10  WklyStudyHours       29686 non-null  object
 11  MathScore            30641 non-null  int64
 12  ReadingScore         30641 non-null  int64
 13  WritingScore         30641 non-null  int64
dtypes: float64(1), int64(3), object(10)
memory usage: 3.3+ MB
```

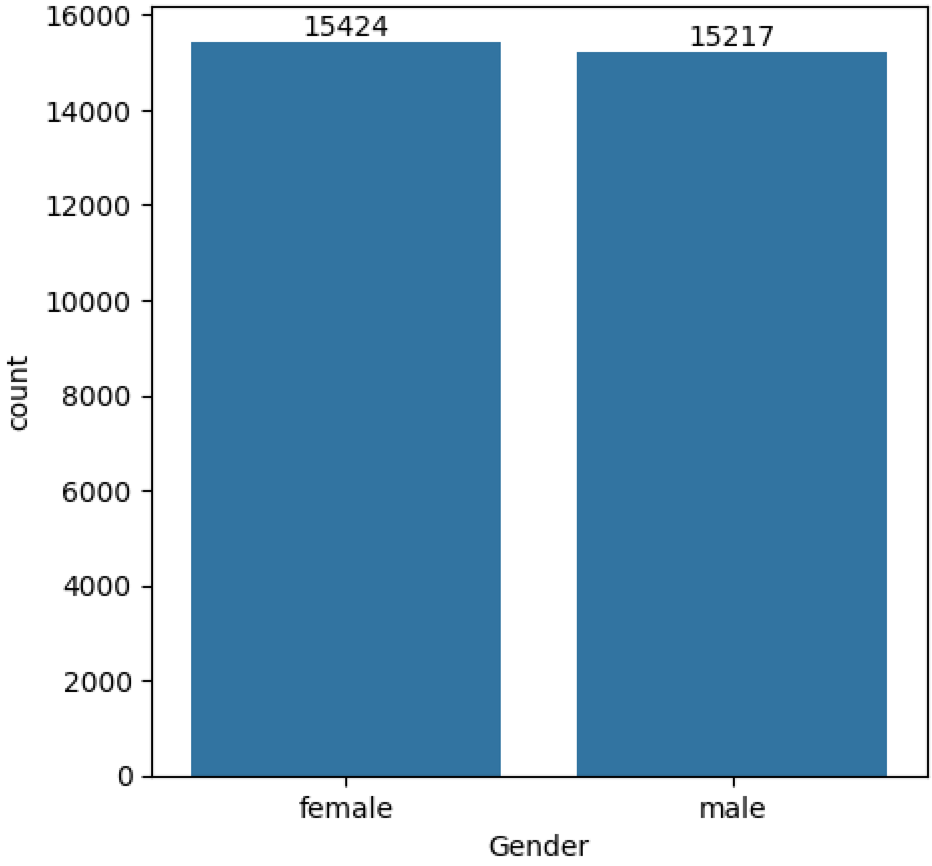
Gender Distribution

In [21]:

```
plt.figure(figsize = (5,5))
ax = sns.countplot(data = df, x = "Gender")
ax.bar_label(ax.containers[0])
plt.show
```

Out[21]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



from the above chat we have analysed that the number of number of females in the data is more than the number of males

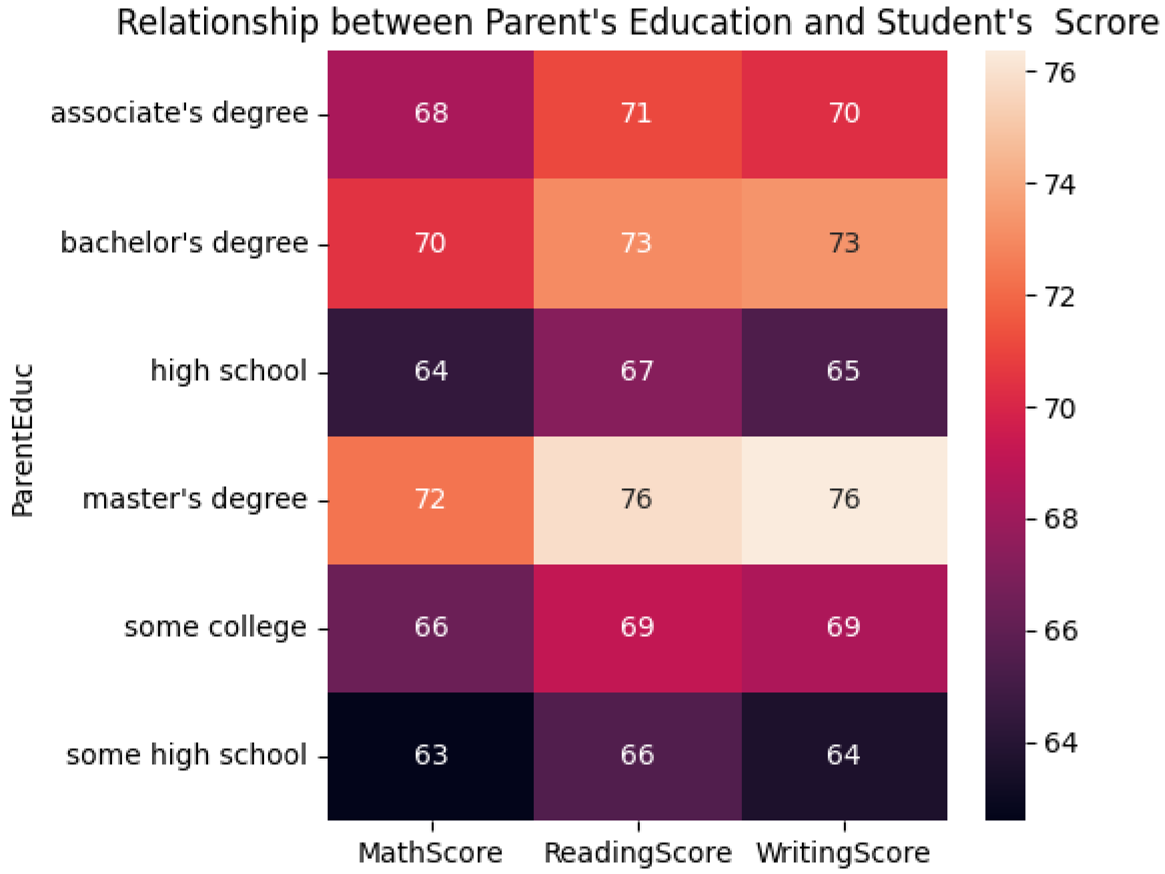
In [27]:

```
gb = df.groupby("ParentEduc").agg({"MathScore":'mean',"ReadingScore":'mean', "WritingSco
print(gb)
```

```
                    MathScore  ReadingScore  WritingScore
ParentEduc
associate's degree  68.365586     71.124324     70.299099
bachelor's degree   70.466627     73.062020     73.331069
high school         64.435731     67.213997     65.421136
master's degree     72.336134     75.832921     76.356896
some college        66.390472     69.179708     68.501432
some high school    62.584013     65.510785     63.632409
```

In [35]:

```
plt.figure(figsize = (5,5))
plt.title("Relationship between Parent's Education and Student's  Scrore")
sns.heatmap(gb, annot=True)
plt.show()
```



from the above chart we have concluded that the education of the parents have a good impact on the student performance

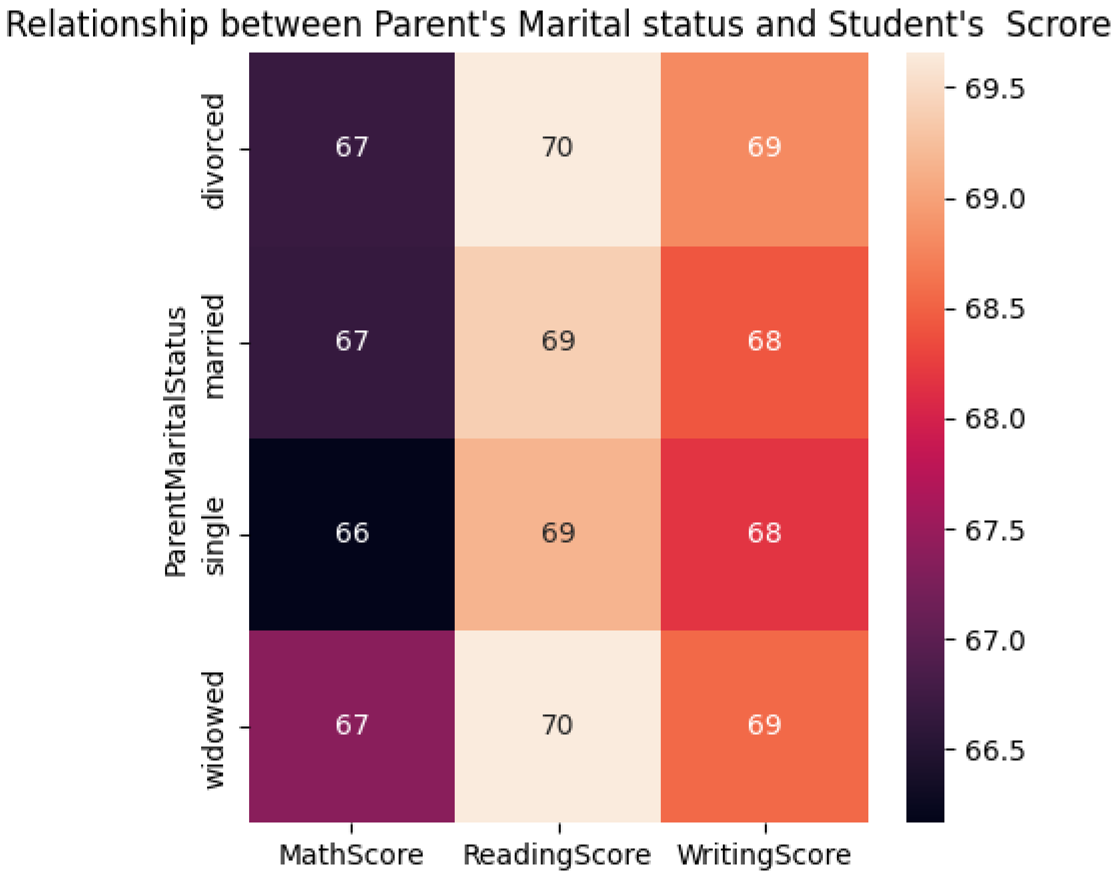
In [32]:

```
gb1 = df.groupby("ParentMaritalStatus").agg({"MathScore":'mean',"ReadingScore":'mean', "
print(gb1)
```

```
                     MathScore  ReadingScore  WritingScore
ParentMaritalStatus
divorced             66.691197     69.655011     68.799146
married              66.657326     69.389575     68.420981
single               66.165704     69.157250     68.174440
widowed              67.368866     69.651438     68.563452
```

In [36]:

```
plt.figure(figsize = (5,5))
plt.title("Relationship between Parent's Marital status and Student's  Scrore")
sns.heatmap(gb1, annot=True)
plt.show()
```



from the above chart we have concluded that the marital status of the parents have negligible impact on the student performance

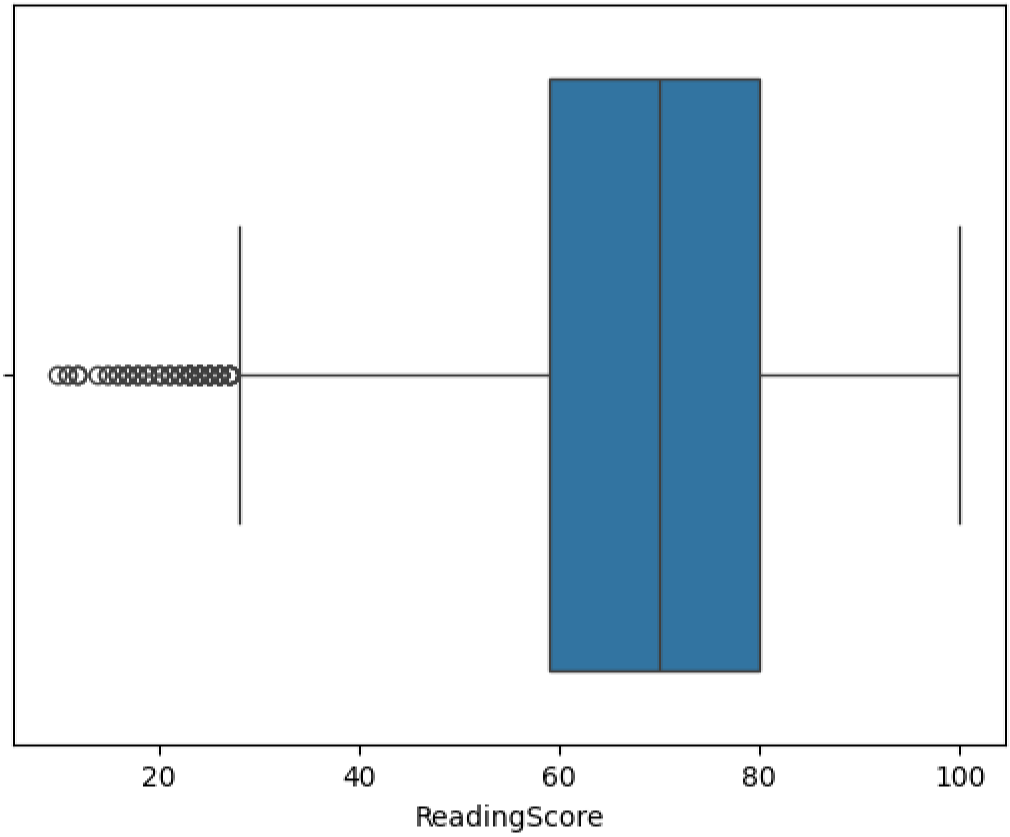
In [37]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30641 entries, 0 to 30640
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Gender               30641 non-null  object 
 1   EthnicGroup          28801 non-null  object 
 2   ParentEduc           28796 non-null  object 
 3   LunchType            30641 non-null  object 
 4   TestPrep             28811 non-null  object 
 5   ParentMaritalStatus  29451 non-null  object 
 6   PracticeSport        30010 non-null  object 
 7   IsFirstChild         29737 non-null  object 
 8   NrSiblings           29069 non-null  float64
 9   TransportMeans       27507 non-null  object 
 10  WklyStudyHours       29686 non-null  object 
 11  MathScore            30641 non-null  int64  
 12  ReadingScore         30641 non-null  int64  
 13  WritingScore         30641 non-null  int64
dtypes: float64(1), int64(3), object(10)
memory usage: 3.3+ MB
```

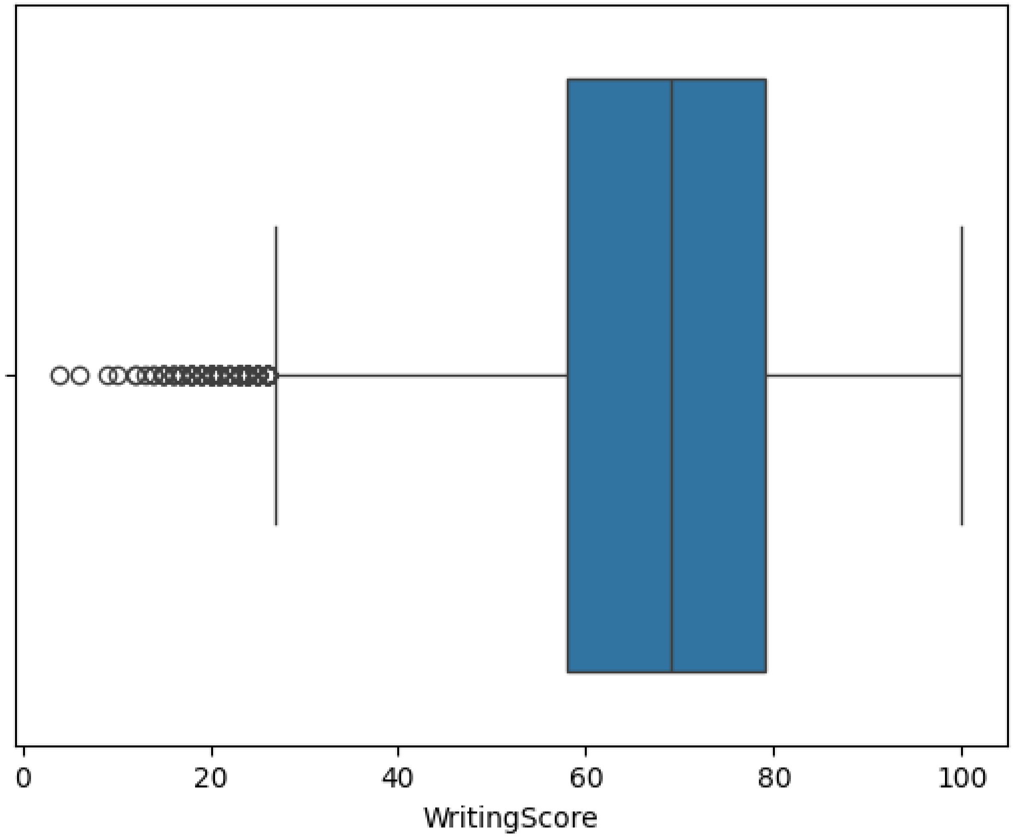
In [52]:

```
sns.boxplot(data = df, x = "ReadingScore")
plt.show()
```



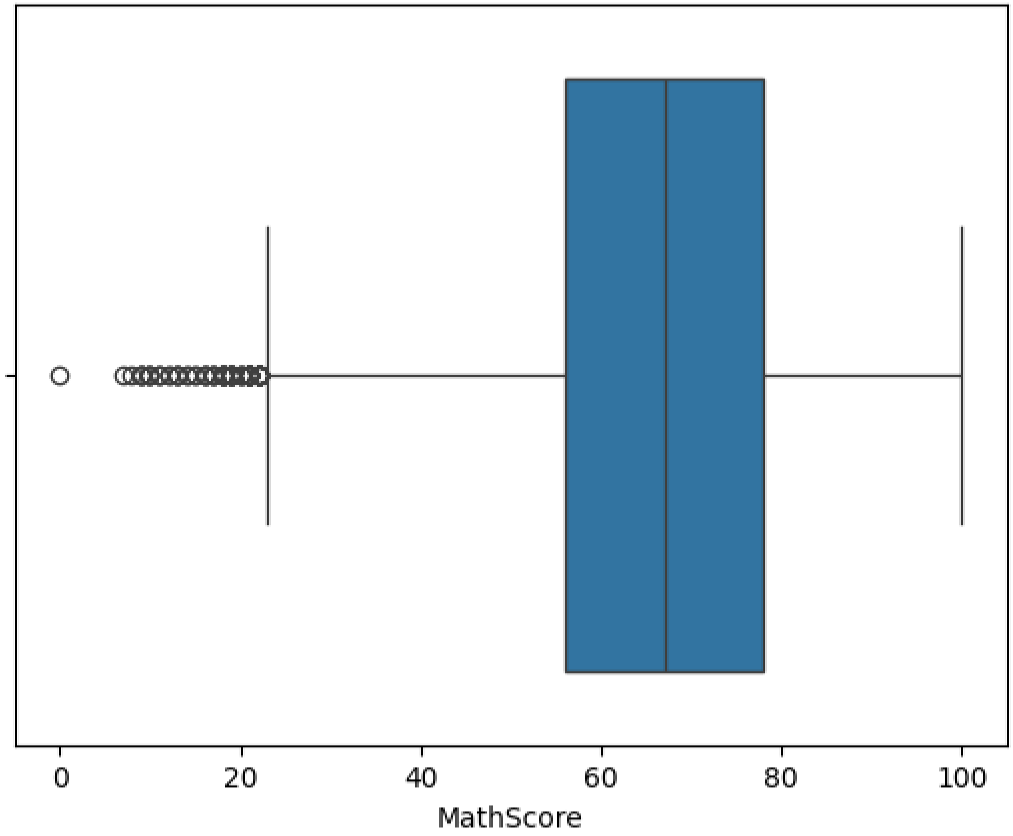
In [53]:

```
sns.boxplot(data = df, x = "WritingScore")
plt.show()
```



In [54]:

```
sns.boxplot(data = df, x = "MathScore")
plt.show()
```

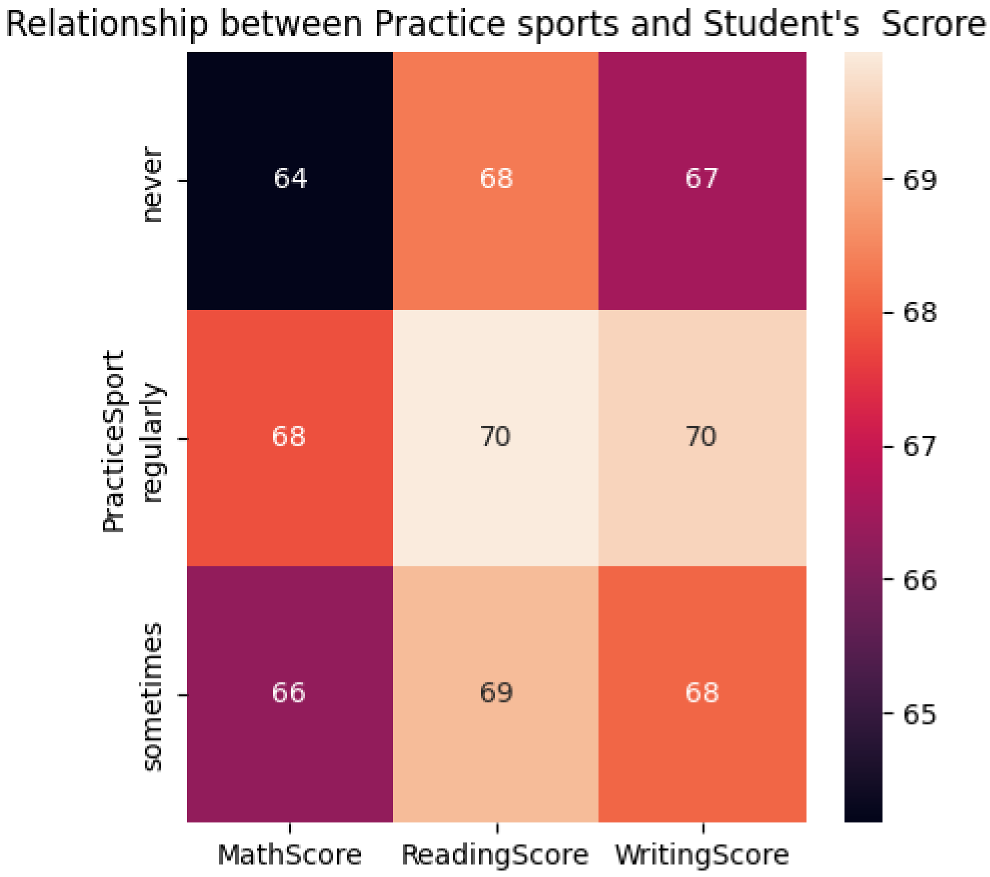


means that maths has the very difficult subject compare to the reading and writhing

In [38]:

```
gb2 = df.groupby("PracticeSport").agg({"MathScore":'mean',"ReadingScore":'mean', "Writin
print(gb2)
plt.figure(figsize = (5,5))
plt.title("Relationship between Practice sports and Student's  Scrore")
sns.heatmap(gb2, annot=True)
plt.show()
```

```
               MathScore  ReadingScore  WritingScore
PracticeSport                                        
never          64.171079     68.337662     66.522727
regularly      67.839155     69.943019     69.604003
sometimes      66.274831     69.241307     68.072438
```



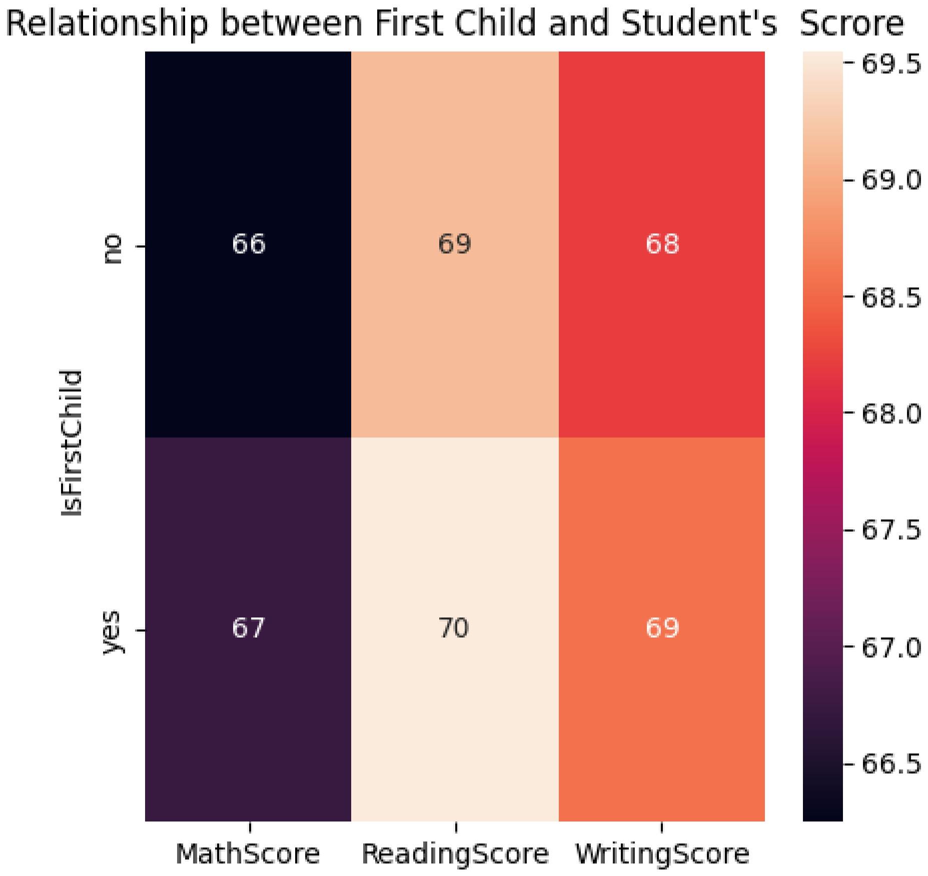
from the above chart we have concluded that the Practicing sports regularly has a slightly impact on the student performance

In [40]:

```
gb3 = df.groupby("IsFirstChild").agg({"MathScore":'mean',"ReadingScore":'mean', "Writing
print(gb3)
plt.figure(figsize = (5,5))
plt.title("Relationship between First Child and Student's  Scrore")
```

```
sns.heatmap(gb3, annot=True)
plt.show()
```

```
              MathScore  ReadingScore  WritingScore
IsFirstChild
no            66.246832     69.132614     68.210887
yes           66.740646     69.542553     68.558484
```

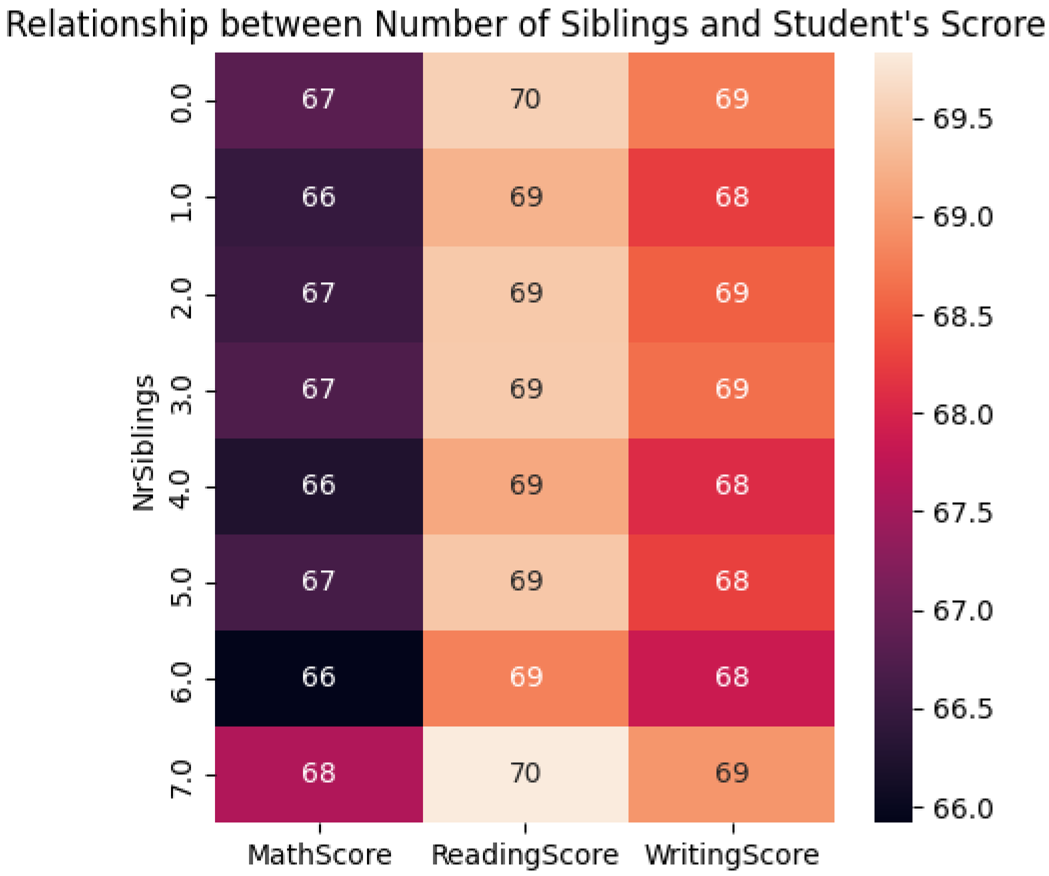


from the above chart we have concluded that the being a first child has not significant impact on the student performance

In [41]:

```
gb4 = df.groupby("NrSiblings").agg({"MathScore":'mean',"ReadingScore":'mean', "WritingSc
print(gb4)
plt.figure(figsize = (5,5))
plt.title("Relationship between Number of Siblings and Student's Scrore")
sns.heatmap(gb4, annot=True)
plt.show()
```

```
            MathScore  ReadingScore  WritingScore
NrSiblings
0.0         66.819449     69.547812     68.746515
1.0         66.473896     69.259097     68.245345
2.0         66.554934     69.472018     68.522533
3.0         66.719092     69.488159     68.650498
4.0         66.245495     69.144169     68.073444
5.0         66.630303     69.453788     68.282576
6.0         65.917219     68.801325     67.860927
7.0         67.615120     69.828179     68.986254
```



from the above chart we have concluded that the number of sibling has not significant impact on the student performance

In [43]:

```
df.info()
```

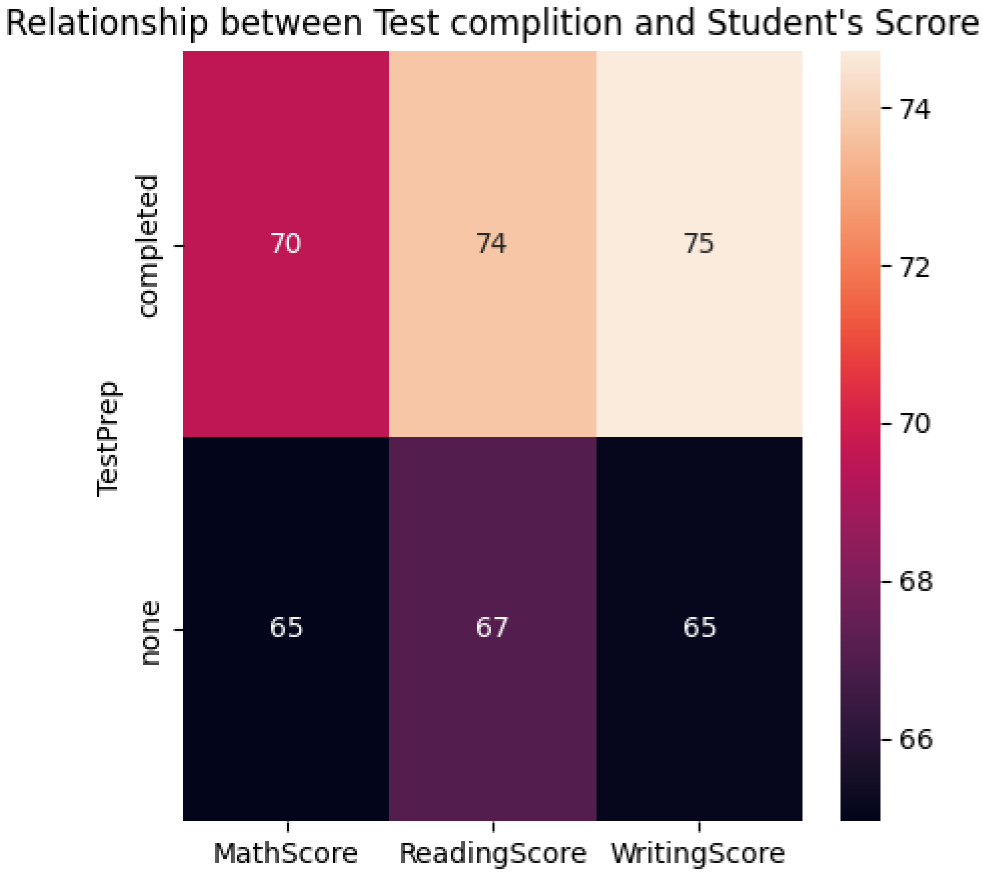
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30641 entries, 0 to 30640
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Gender               30641 non-null  object 
 1   EthnicGroup          28801 non-null  object 
 2   ParentEduc           28796 non-null  object 
 3   LunchType            30641 non-null  object 
 4   TestPrep             28811 non-null  object 
 5   ParentMaritalStatus  29451 non-null  object 
 6   PracticeSport        30010 non-null  object 
 7   IsFirstChild         29737 non-null  object 
 8   NrSiblings           29069 non-null  float64
 9   TransportMeans       27507 non-null  object 
 10  WklyStudyHours       29686 non-null  object 
 11  MathScore            30641 non-null  int64  
 12  ReadingScore         30641 non-null  int64  
 13  WritingScore         30641 non-null  int64  
dtypes: float64(1), int64(3), object(10)
memory usage: 3.3+ MB
```

In [44]:

```
gb5 = df.groupby("TestPrep").agg({"MathScore":'mean',"ReadingScore":'mean', "WritingScor
print(gb5)
```

```
plt.figure(figsize = (5,5))
plt.title("Relationship between Test complition and Student's Scrore")
sns.heatmap(gb5, annot=True)
plt.show()
```

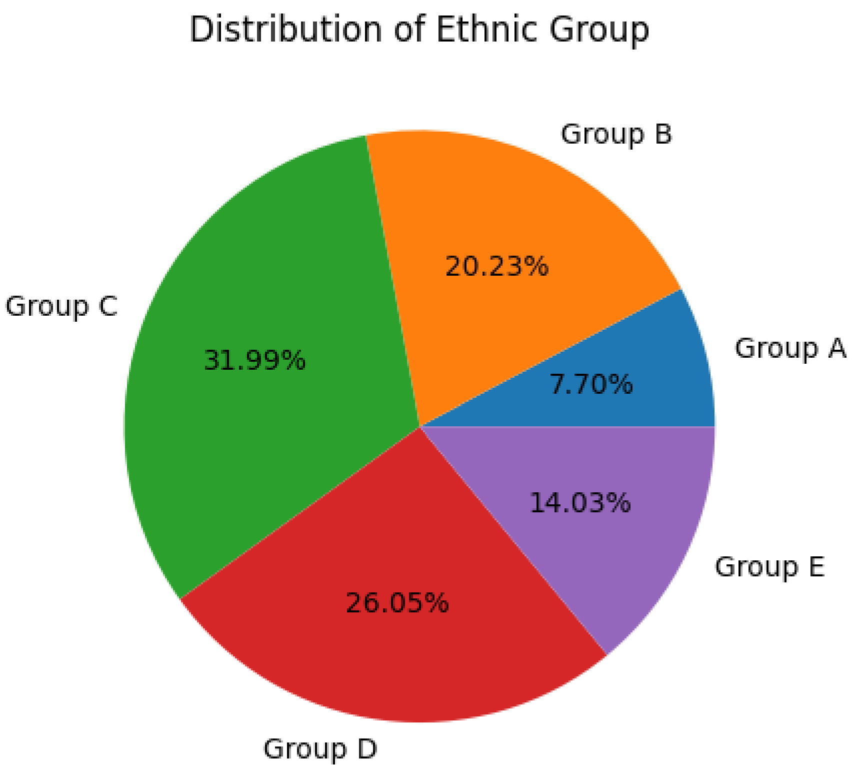
```
           MathScore  ReadingScore  WritingScore
TestPrep
completed   69.54666     73.732998     74.703265
none        64.94877     67.051071     65.092756
```



In [77]:

```
#distribution of Ethnic Groups
groupA = df.loc[(df['EthnicGroup'] == "group A")].count()
groupB = df.loc[(df['EthnicGroup'] == "group B")].count()
groupC = df.loc[(df['EthnicGroup'] == "group C")].count()
groupD = df.loc[(df['EthnicGroup'] == "group D")].count()
groupE = df.loc[(df['EthnicGroup'] == "group E")].count()

l = ["Group A", "Group B", "Group C", "Group D", "Group E"]
mylist = [groupA["EthnicGroup"], groupB["EthnicGroup"], groupC["EthnicGroup"], groupD["E
plt.pie(mylist, labels = l, autopct = "%1.2f%%")
plt.title("Distribution of Ethnic Group")
plt.show()
```



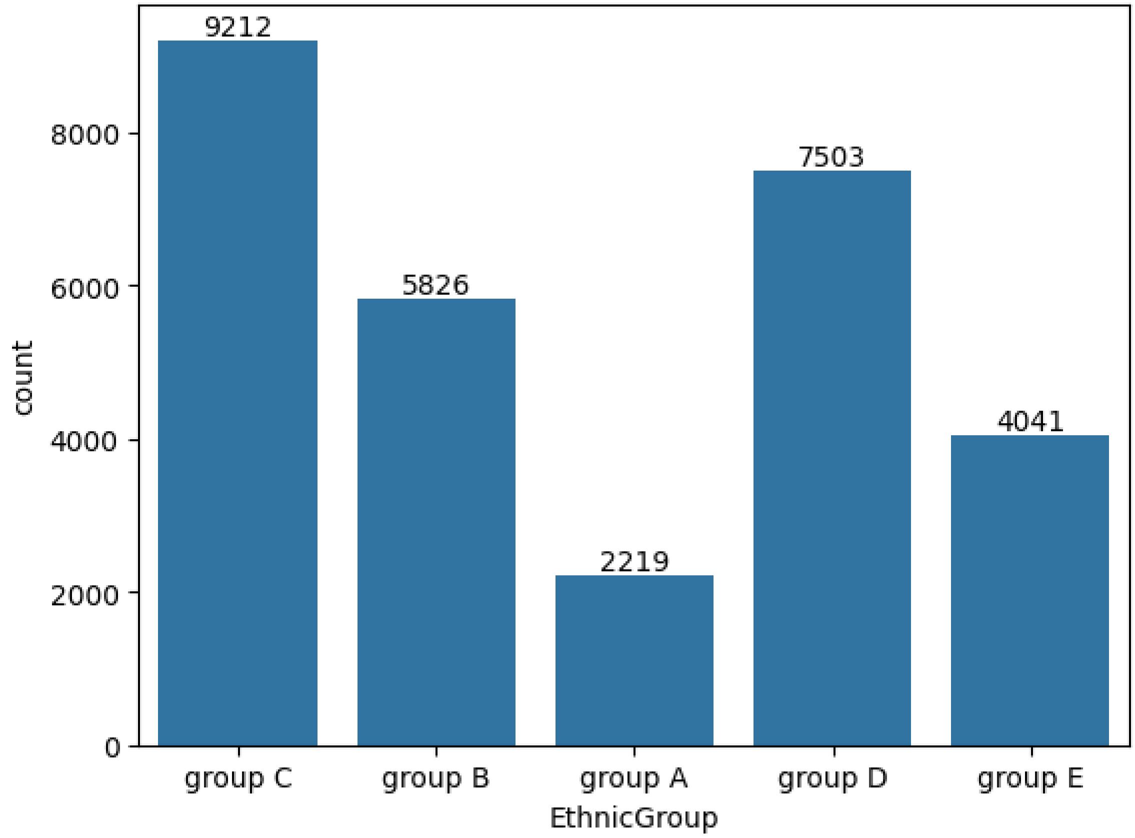
from the above chart we have concluded that the Test complition has a significant impact on the student performance

In [75]:

```
ax = sns.countplot(data = df, x = 'EthnicGroup')
ax.bar_label(ax.containers[0])
```

Out[75]:

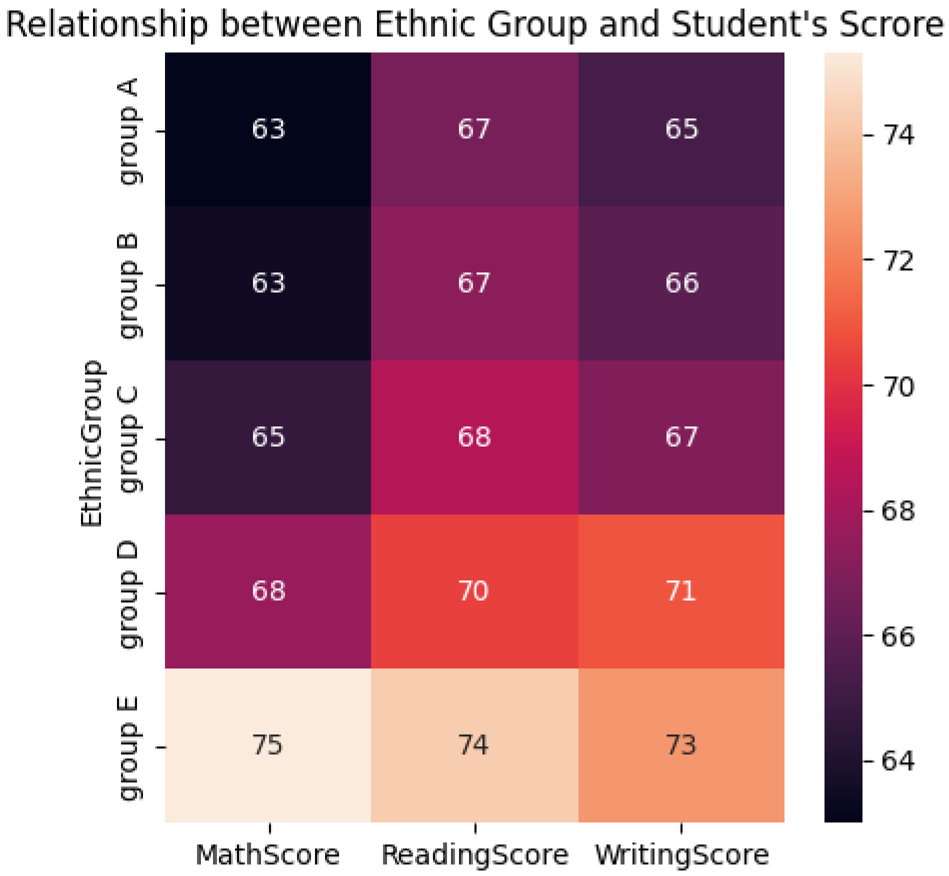
```
[Text(0, 0, '9212'),
 Text(0, 0, '5826'),
 Text(0, 0, '2219'),
 Text(0, 0, '7503'),
 Text(0, 0, '4041')]
```



In [45]:

```
gb6 = df.groupby("EthnicGroup").agg({"MathScore":'mean',"ReadingScore":'mean', "WritingS
print(gb6)
plt.figure(figsize = (5,5))
plt.title("Relationship between Ethnic Group and Student's Scrore")
sns.heatmap(gb6, annot=True)
plt.show()
```

```
             MathScore  ReadingScore  WritingScore
EthnicGroup                                       
group A      62.991888     66.787742     65.251915
group B      63.490216     67.320460     65.895125
group C      64.695723     68.438233     66.999240
group D      67.666400     70.382247     70.890844
group E      75.298936     74.251423     72.677060
```



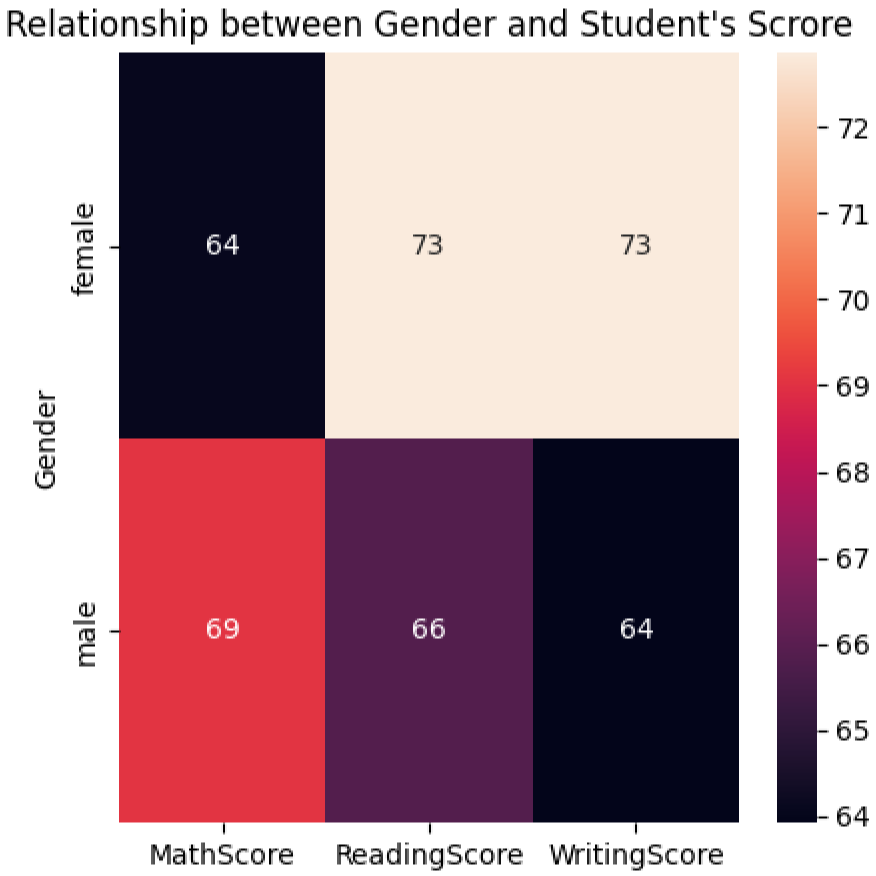
from the above graph the ethnic Group C has the higher count than the others

from the above chart we have concluded that the Ethnic group a significant impact on the student performance Like ethnic groupE's has the higher performace and groupA has lower performac

In [47]:

```
gb7 = df.groupby("Gender").agg({"MathScore":'mean',"ReadingScore":'mean', "WritingScore"
print(gb7)
plt.figure(figsize = (5,5))
plt.title("Relationship between Gender and Student's Scrore")
sns.heatmap(gb7, annot=True)
plt.show()
```

```
        MathScore  ReadingScore  WritingScore
Gender
female  64.080654     72.853216     72.856457
male    69.069856     65.854571     63.920418
```

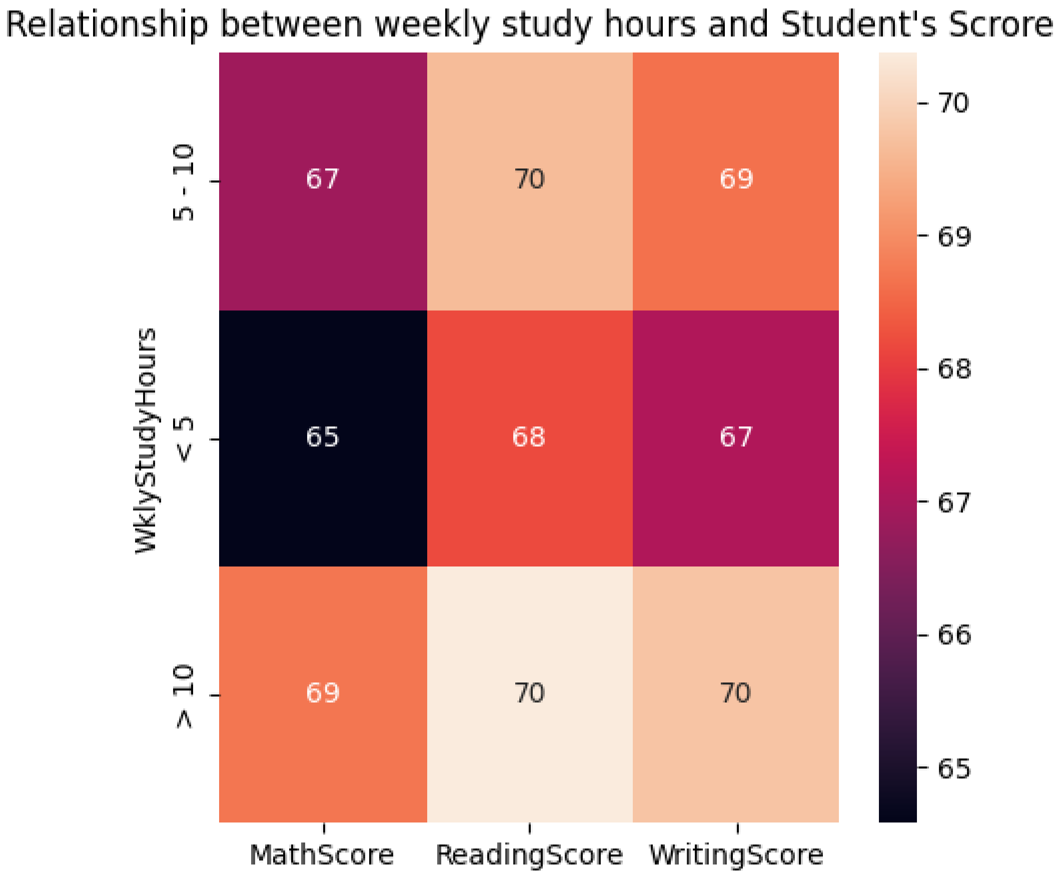


from the above chart we have concluded that the male are good in maths and female are good in reading and writing score

In [48]:

```
gb8 = df.groupby("WklyStudyHours").agg({"MathScore":'mean',"ReadingScore":'mean', "Writi
print(gb8)
plt.figure(figsize = (5,5))
plt.title("Relationship between weekly study hours and Student's Scrore")
sns.heatmap(gb8, annot=True)
plt.show()
```

```
                MathScore  ReadingScore  WritingScore
WklyStudyHours
5 - 10          66.870491     69.660532     68.636280
< 5             64.580359     68.176135     67.090192
> 10            68.696655     70.365436     69.777778
```

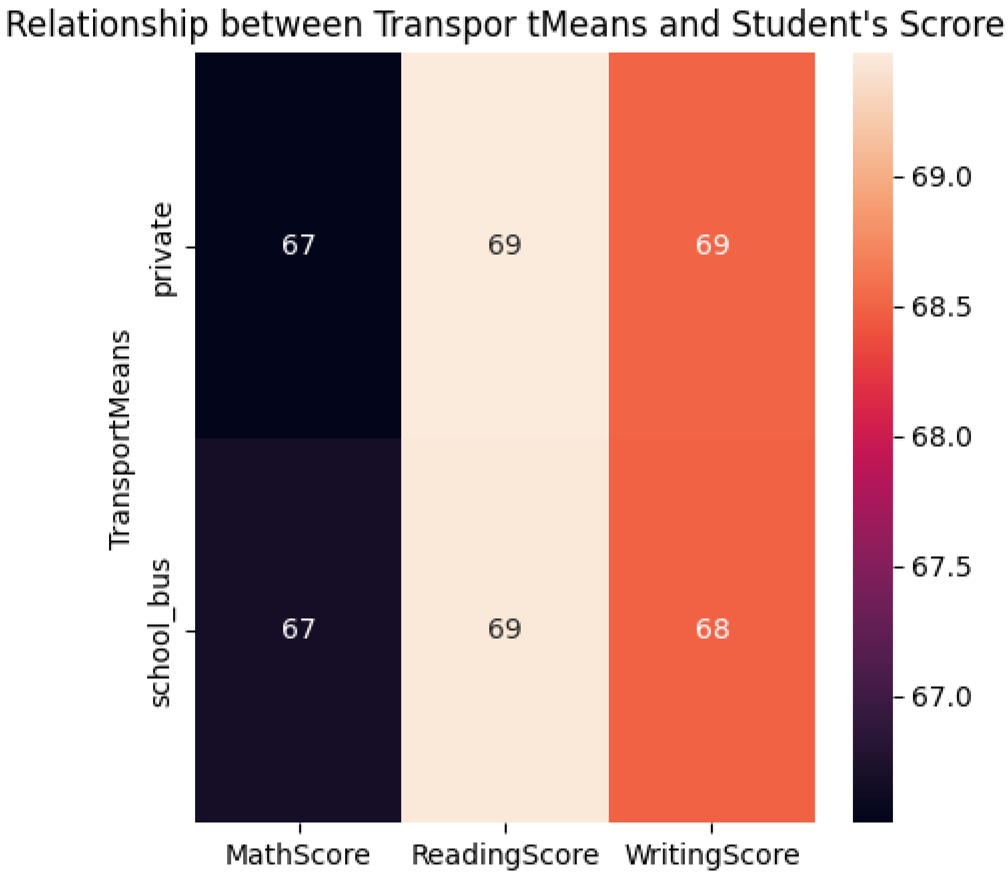


from the above chart we have concluded that the students studied less than 5h have lower score and 5 to 10 and more 10 have the higher score in math, reading and writing score

In [49]:

```
gb9 = df.groupby("TransportMeans").agg({"MathScore":'mean',"ReadingScore":'mean', "Writi
print(gb9)
plt.figure(figsize = (5,5))
plt.title("Relationship between Transpor tMeans and Student's Scrore")
sns.heatmap(gb9, annot=True)
plt.show()
```

```
                MathScore  ReadingScore  WritingScore
TransportMeans
private         66.511354     69.472364     68.509593
school_bus      66.674636     69.446206     68.492351
```

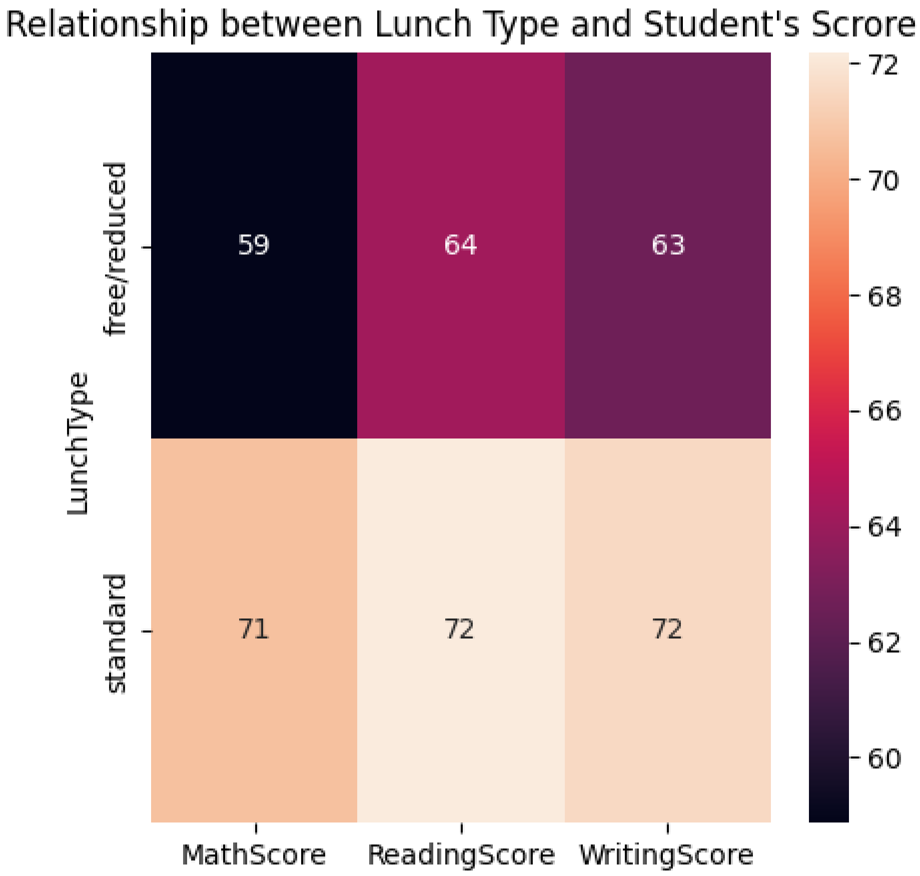


from the above chart we have concluded that the students Transport means has no impact on the student score

In [50]:

```
gb10 = df.groupby("LunchType").agg({"MathScore":'mean',"ReadingScore":'mean', "WritingSc
print(gb10)
plt.figure(figsize = (5,5))
plt.title("Relationship between Lunch Type and Student's Scrore")
sns.heatmap(gb10, annot=True)
plt.show()
```

```
              MathScore  ReadingScore  WritingScore
LunchType
free/reduced  58.862332     64.189735     62.650522
standard      70.709370     72.175634     71.529716
```



from the above chart we have concluded that the student's Lunch type plays a significate role in the student's performace because free or reduced lunchtype negativly impacts to the student's score though good or standard meal can contribte to the student's higher performace

```
In [ ]:
```