In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df= pd.read_csv("students_data.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         10-May         69            90            88
2            < 5         87            93            91
3         10-May         45            56            42
4         10-May         76            78            75
```

In [5]:
```
df.describe()
```

Out[5]:

	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 [6]:
```
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 [7]:
```
df.isnull().sum()
```

Out[7]:
```
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
```

In [8]:
```
df= df.drop("Unnamed: 0", axis=1)
print (df.head())
```

```
  Gender EthnicGroup          ParentEduc     LunchType TestPrep  \
0  female         NaN   bachelor's degree      standard     none
1  female     group C        some college      standard      NaN
2  female     group B     master's degree      standard     none
3    male     group A  associate's degree  free/reduced     none
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         10-May         69            90            88
2            < 5         87            93            91
3         10-May         45            56            42
4         10-May         76            78            75
```

In [9]:
```
df["WklyStudyHours"]=df["WklyStudyHours"].str.replace("10-May","5-10")
df.head()
```

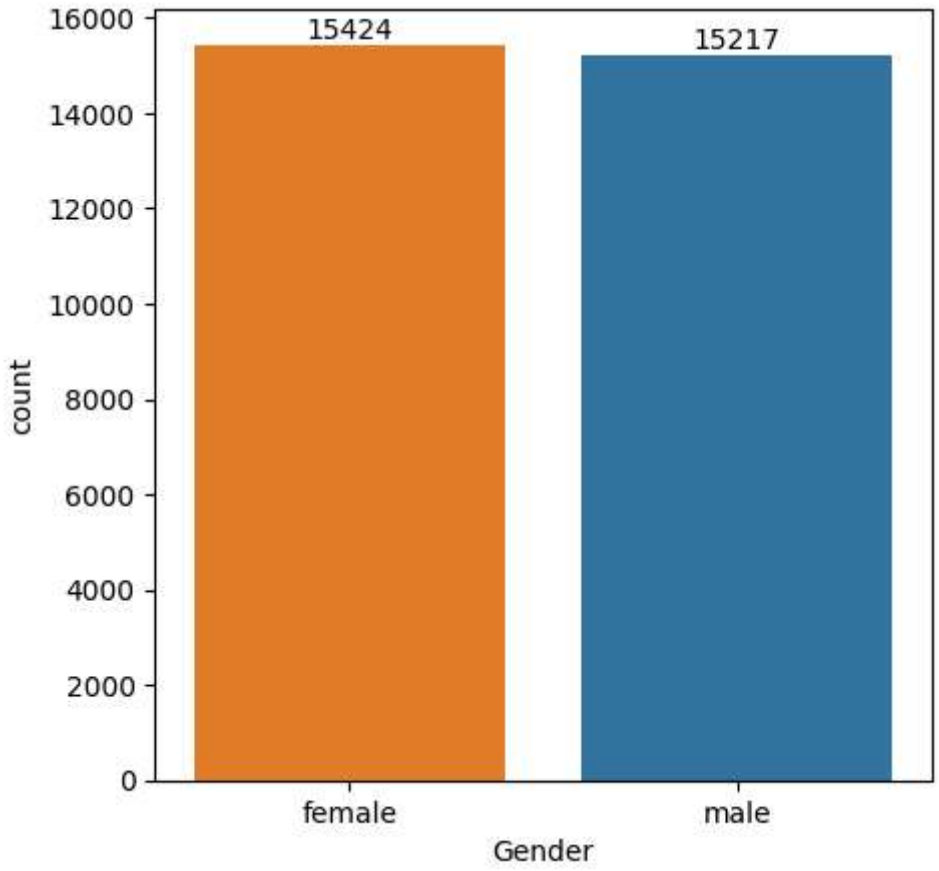
Out[9]:

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeS
0	female	NaN	bachelor's degree	standard	none	married	regu
1	female	group C	some college	standard	NaN	married	somet
2	female	group B	master's degree	standard	none	single	somet
3	male	group A	associate's degree	free/reduced	none	married	r
4	male	group C	some college	standard	none	married	somet

In [10]:
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(5,5))

# Custom colors for Male & Female
ax = sns.countplot(
    data=df,
    x="Gender",
    palette={"male": "#1f77b4", "female": "#ff7f0e"}  # Blue & Orange
)

# Har bar ke upar label dikhane ke liye
for container in ax.containers:
```

```
    ax.bar_label(container)

plt.show()
```

```
C:\Users\RK PCS\AppData\Local\Temp\ipykernel_8156\2724906359.py:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  ax = sns.countplot(
```



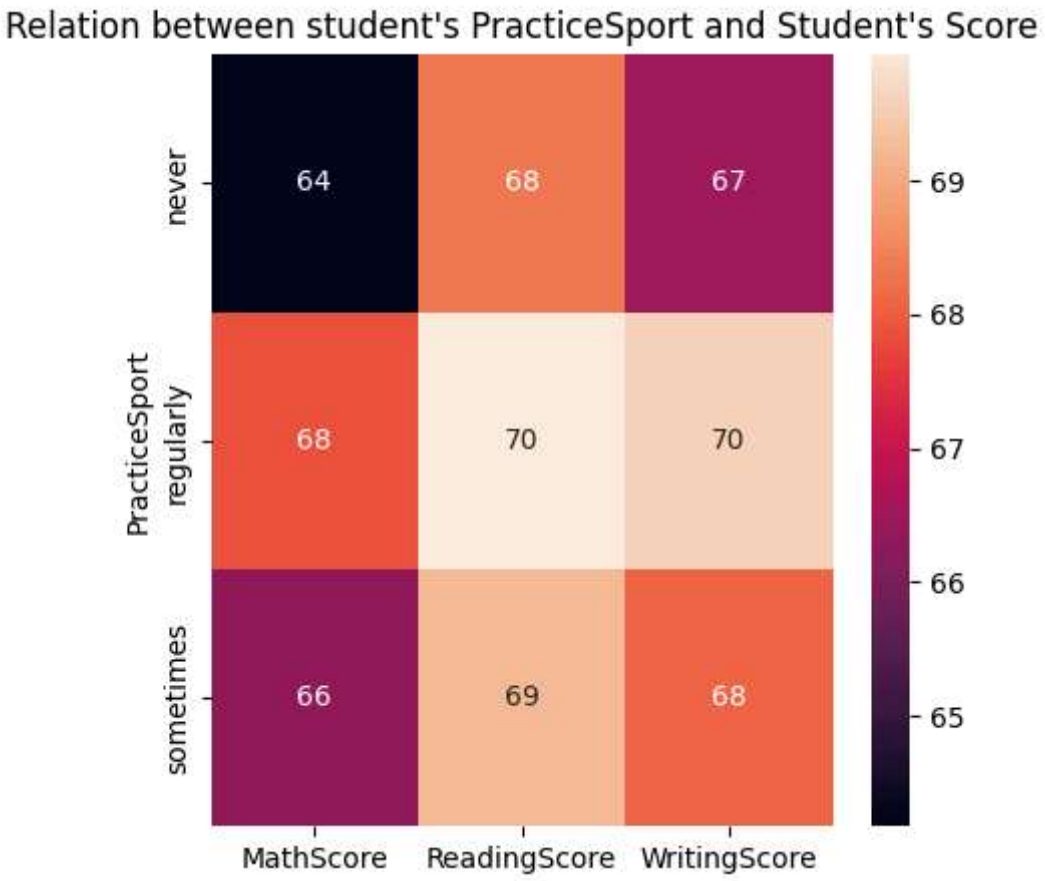
from the above Chart we have analyzed that the no female students are more than male

In [11]:
```
gb= df.groupby("PracticeSport").agg({"MathScore":'mean',"ReadingScore":'mean',"Writ
print(gb)
```

```
               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
```

In [12]:
```
plt.figure(figsize = (5,5))
sns.heatmap(gb, annot= True)
```

```
plt.title("Relation between student's PracticeSport and Student's Score")
plt.show()
```

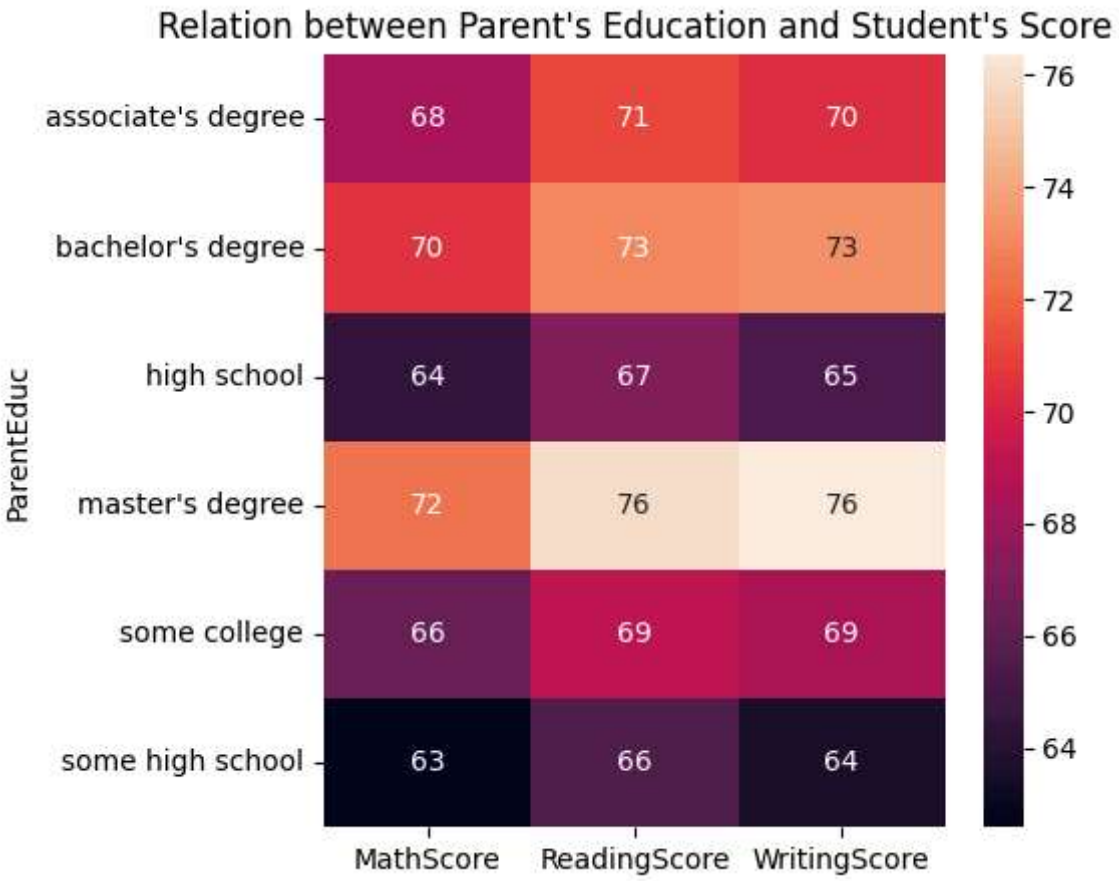


It is concluded that practicing sports effect students scores

In [17]:
```
gb1= df.groupby("ParentEduc").agg({"MathScore":'mean',"ReadingScore":'mean',"Writin
print(gb1)
```

```
                   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 [20]:
```
plt.figure(figsize = (5,5))
sns.heatmap(gb1, annot= True)
plt.title("Relation between Parent's Education and Student's Score")
plt.show()
```

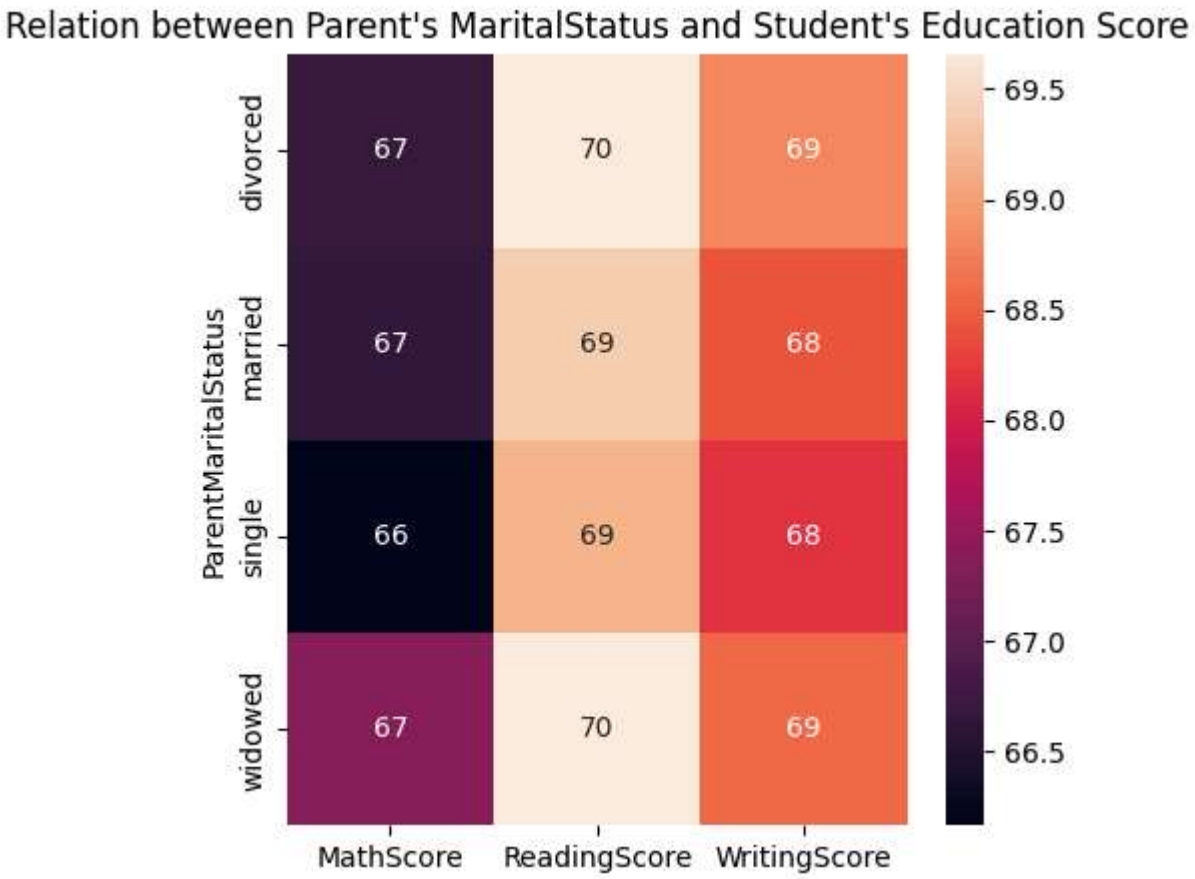


It is concluded from above chart and data Parent's Education effects Students Scores

In [21]:
```
gb2= df.groupby("ParentMaritalStatus").agg({"MathScore":'mean',"ReadingScore":'mean
print(gb2)
```

```
                    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 [22]:
```
plt.figure(figsize = (5,5))
sns.heatmap(gb2, annot= True)
plt.title("Relation between Parent's MaritalStatus and Student's Education Score")
plt.show()
```

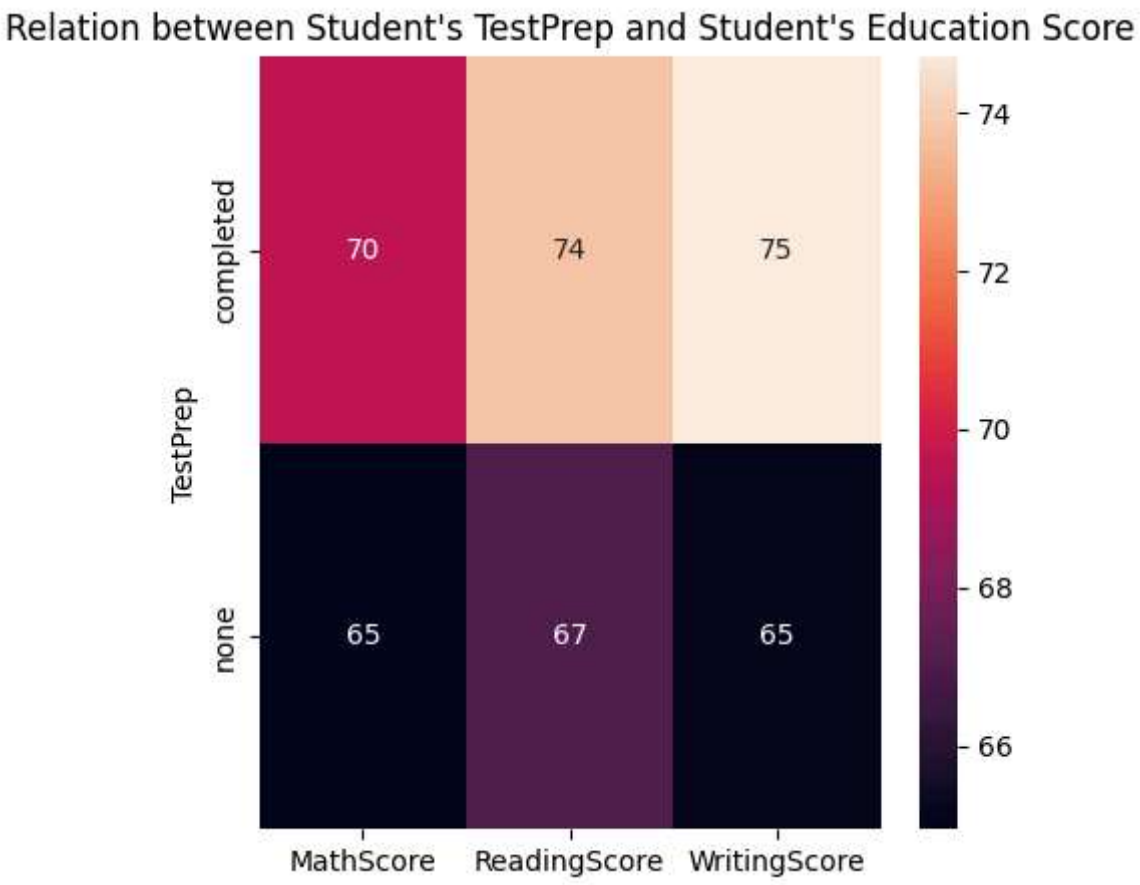


it is concluded that the marital status does not effect that much/negligibile on student's score

In [24]:
```
gb3= df.groupby("TestPrep").agg({"MathScore":'mean',"ReadingScore":'mean',"WritingS
print(gb3)
```

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

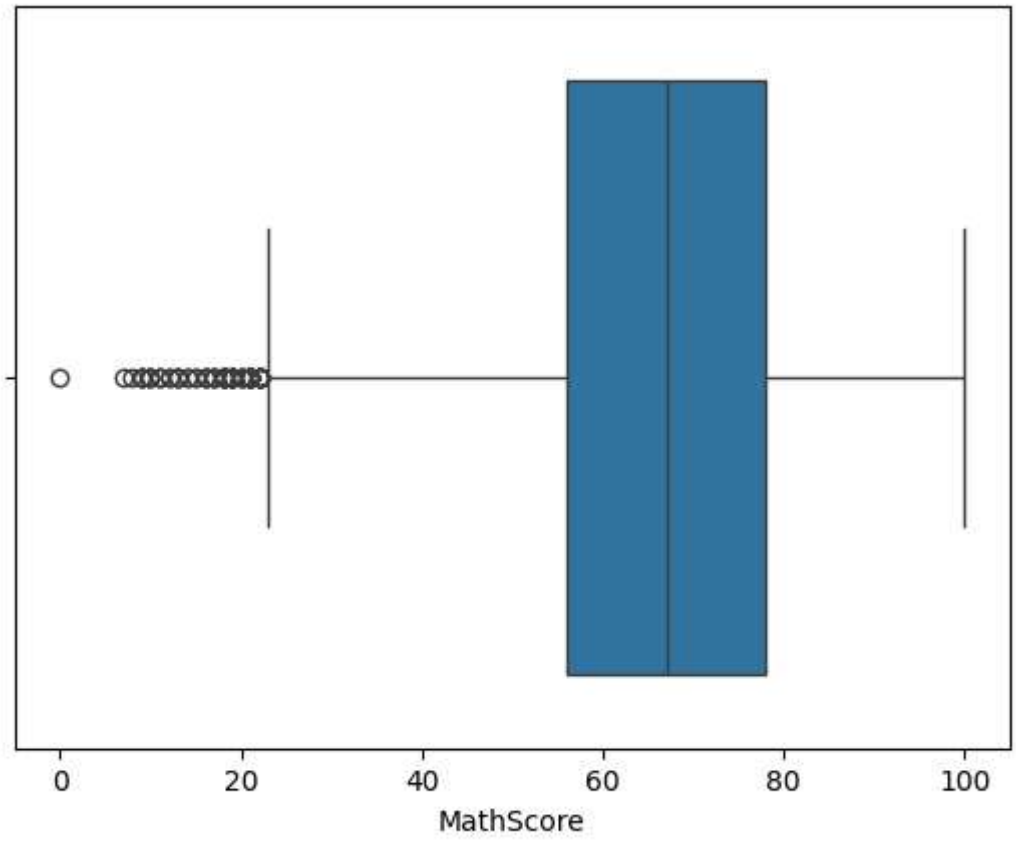
In [25]:
```
plt.figure(figsize = (5,5))
sns.heatmap(gb3, annot= True)
plt.title("Relation between Student's TestPrep and Student's Education Score")
plt.show()
```



It is concluded that students practicing more for studies get higher marks

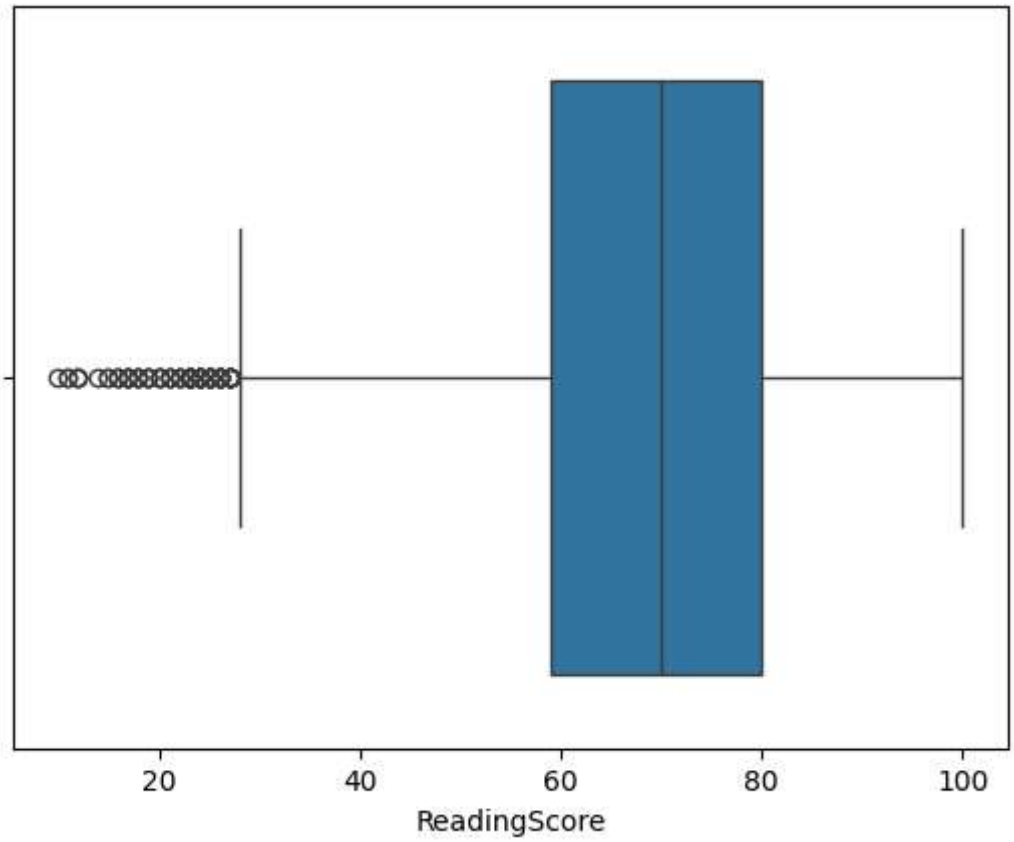
In [26]:
```
sns.boxplot(data = df , x = "MathScore")
plt.show
```

Out[26]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



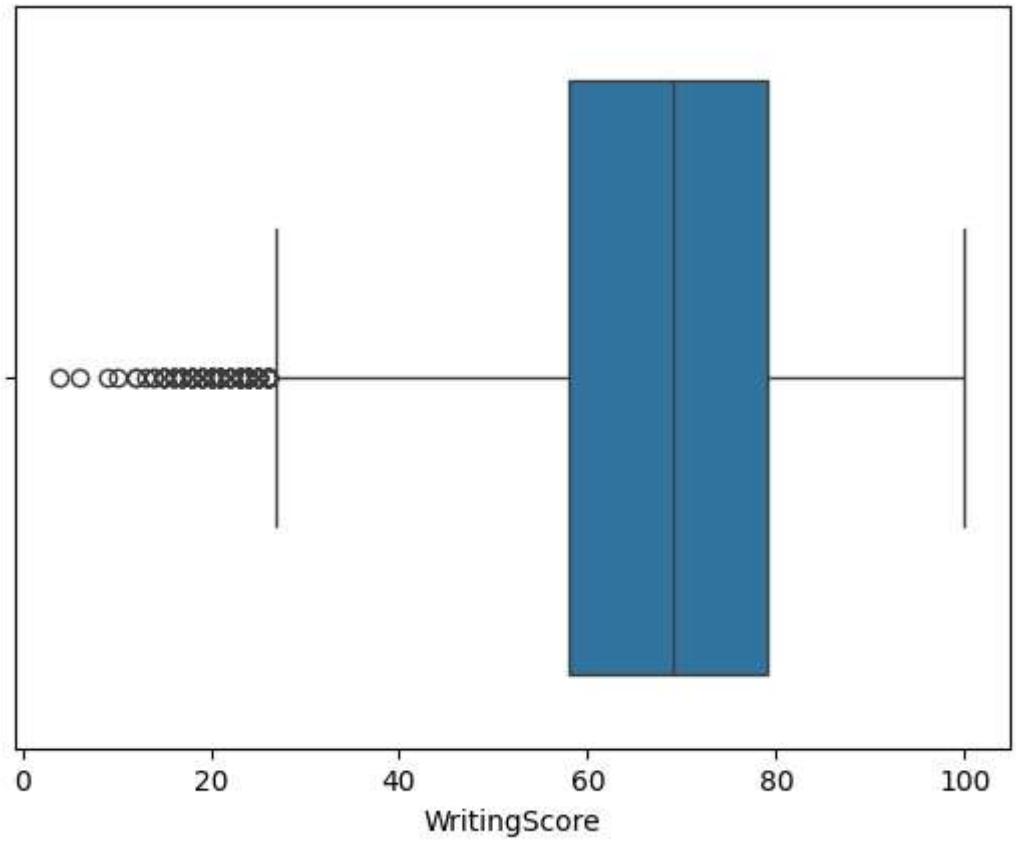
In [27]:
```
sns.boxplot(data = df , x = "ReadingScore")
plt.show
```

Out[27]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [28]:
```
sns.boxplot(data = df , x = "WritingScore")
plt.show
```

Out[28]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [29]:
```
print (df["EthnicGroup"].unique())
```

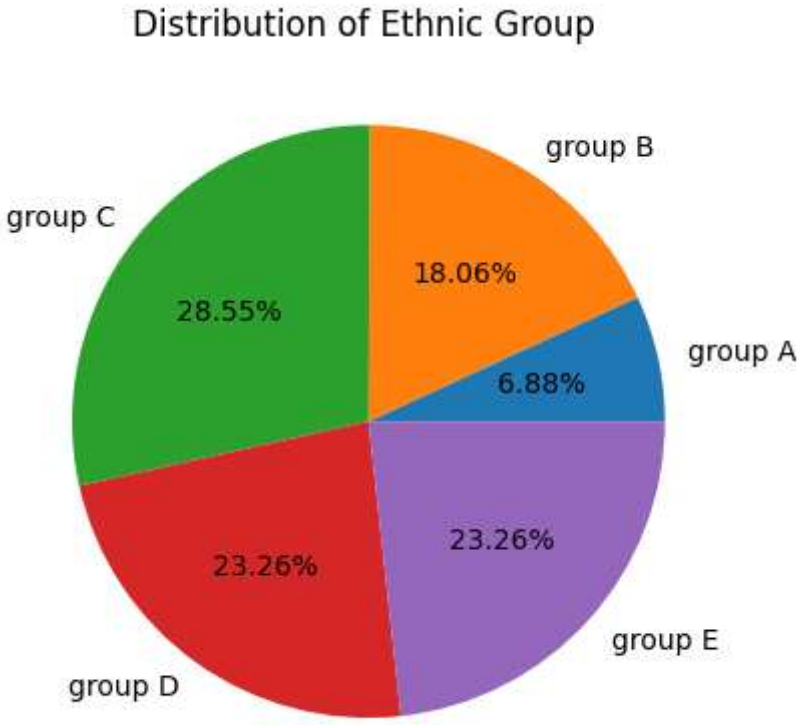
```
[nan 'group C' 'group B' 'group A' 'group D' 'group E']
```

In [30]:
```
groupA= df.loc[(df['EthnicGroup'] == "group A")].count()
print (groupA)
```

```
Gender                 2219
EthnicGroup            2219
ParentEduc             2078
LunchType              2219
TestPrep               2081
ParentMaritalStatus    2121
PracticeSport          2167
IsFirstChild           2168
NrSiblings             2096
TransportMeans         1999
WklyStudyHours         2146
MathScore              2219
ReadingScore           2219
WritingScore           2219
dtype: int64
```

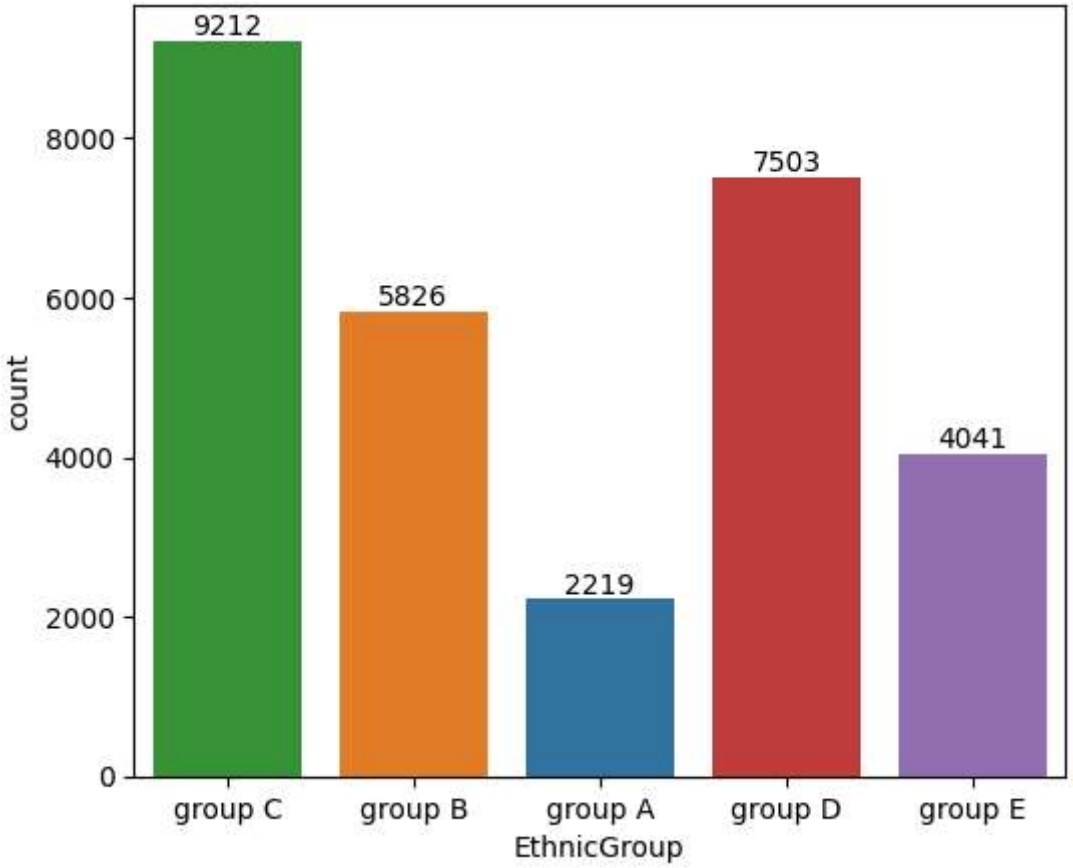
In [26]:
```
groupA= df.loc[(df['EthnicGroup'] == "group A")].count()
groupB= df.loc[(df['EthnicGroup'] == "group B")].count()
groupC= df.loc[(df['EthnicGroup'] == "group C")].count()
groupD= df.loc[(df['EthnicGroup'] == "group D")].count()
groupE= df.loc[(df['EthnicGroup'] == "group E")].count()
l = ["group A","group B","group C","group D","group E"]
mlist=[groupA["EthnicGroup"],groupB["EthnicGroup"],groupC["EthnicGroup"],groupD["Et
```

```
plt.pie(mlist, labels= l, autopct= "%1.2f%%")
plt.title("Distribution of Ethnic Group ")
plt.show()
```



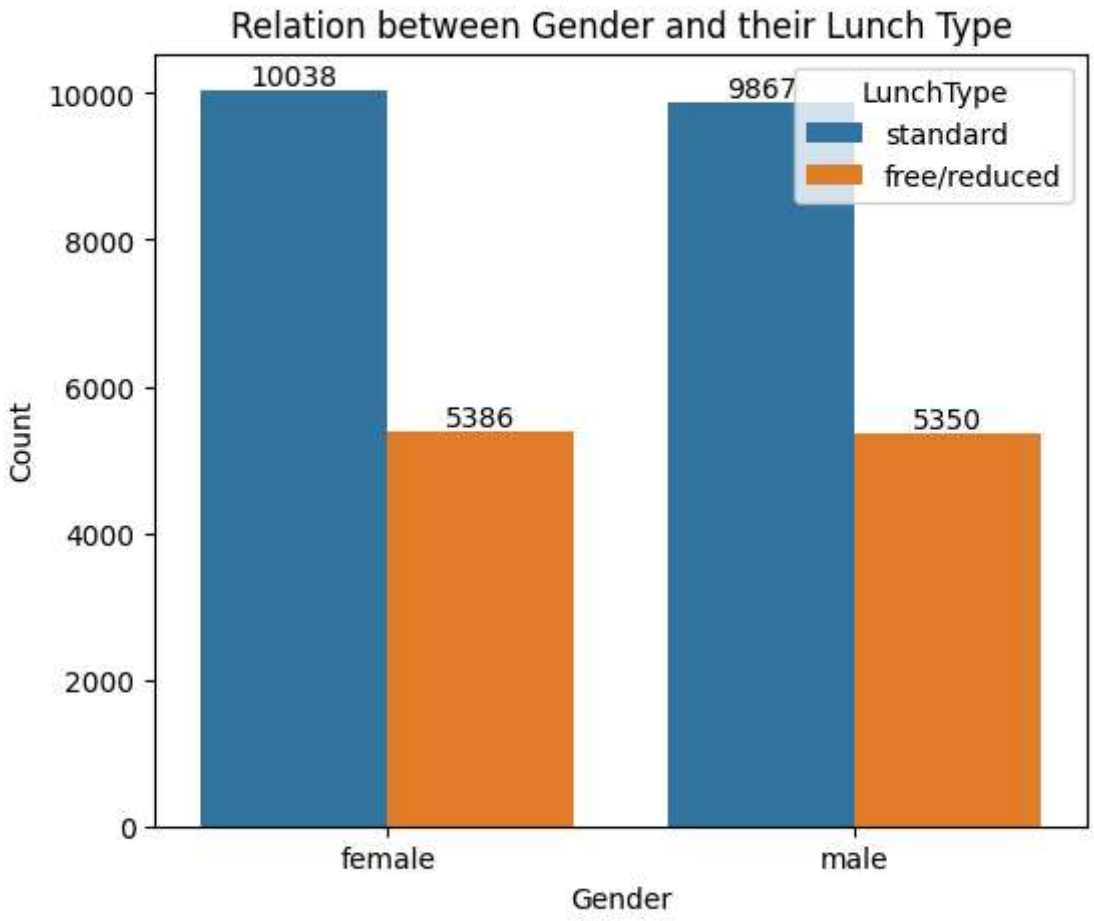
In [31]:
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(6,5))

ax = sns.countplot(
    data=df,
    x='EthnicGroup',
    palette={
        "group A": "#1f77b4",   # Blue
        "group B": "#ff7f0e",   # Orange
        "group C": "#2ca02c",   # Green
        "group D": "#d62728",   # Red
        "group E": "#9467bd"    # Purple
    }
)

for container in ax.containers:
    ax.bar_label(container)

plt.show()
```

```
C:\Users\RK PCS\AppData\Local\Temp\ipykernel_8156\732129671.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  ax = sns.countplot(
```



In [32]:
```
data = {
    'Gender': ['Male', 'Female'],
    'LunchType': ['Standard', 'Free/Reduced']
}
```

In [34]:
```
plt.figure(figsize=(6,5))
ax = sns.countplot(data=df, x='Gender', hue='LunchType')

# Add labels on bars
for container in ax.containers:
    ax.bar_label(container)

plt.title("Relation between Gender and their Lunch Type")
plt.xlabel("Gender")
plt.ylabel("Count")
plt.show()
```


Relation between Gender and their Lunch Type
10038
9867
5386
5350
LunchType
standard
free/reduced
10000
8000
6000
4000
2000
0
Count
female
male
Gender

In [ ]: