In [1]:

```
#Project: Student performance
#we gone analyze a data set for students in a school and we will find some interesti
#This is the link for the data set
'https://www.kaggle.com/spscientist/students-performance-in-exams'
```

Out[1]:

```
'https://www.kaggle.com/spscientist/students-performance-in-exams'
```

In [2]:

```
#Here we gone Install the libraries
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
%matplotlib inline
```

In [3]:

```
# installing the data
df = pd.read_csv('D:\Data sets\StudentsPerformance.csv')
```

In [4]:

```
# displaying the first five lines from the data
df.head()
```

Out[4]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

In [5]:

```
# this code show us the type of data that we working on
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   gender                       1000 non-null   object
 1   race/ethnicity               1000 non-null   object
 2   parental level of education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test preparation course      1000 non-null   object
 5   math score                   1000 non-null   int64
 6   reading score                1000 non-null   int64
 7   writing score                1000 non-null   int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

In [6]:

```
# we have to make sure there is no missing data
df.isnull().sum()
```

Out[6]:

```
gender                         0
race/ethnicity                 0
parental level of education    0
lunch                          0
test preparation course        0
math score                     0
reading score                  0
```

```
writing score                0
dtype: int64
```

In [7]:

```
# we have to make sure there is no duplicated Information among the data set
df.duplicated().sum()
```

Out[7]:

```
0
```

In [8]:

```
# here we can see the lowest and hgihset and average of numbers
df.describe()
```

Out[8]:

	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

In [9]:

```
# this will show to us the names of the columns
df.columns
```

Out[9]:

```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

In [10]:

```
# in this command we will change a varible name so we can use it in the future
df.rename(columns = {'parental level of education':'parental_level_of_education'}, i
```

In [11]:

```
# this code counts the number of males and females
df['gender'].value_counts()
```

Out[11]:

```
female    518
male      482
Name: gender, dtype: int64
```

In [12]:

```
# in this command we will change a varible name so we can use it in the future
df.rename(columns = {'test preparation course':'test_preparation_course'}, inplace =
```

In [13]:

```
# here we can see the males or females who have completed the test preparation cours
pd.crosstab(df.gender, df.test_preparation_course)
```

Out[13]:

test_preparation_course	completed	none
gender		
female	184	334
male	174	308

In [14]:

```
# a code displaying how many students selected standard lunch or free/reduced
pd.crosstab(df.gender, df.lunch)
```

Out[14]:

lunch	free/reduced	standard
gender		
female	189	329
male	166	316

In [15]:

```
# a code showing us the types of lunch and the amount of each one
df['lunch'].value_counts()
```

Out[15]:

```
standard        645
free/reduced    355
Name: lunch, dtype: int64
```

In [16]:

```
group_df = df.groupby("gender")
mean_df = group_df.mean()
```

In [17]:

```
mean_df = mean_df.reset_index()
```

In [18]:

```
#table showing us the average grades of the three subjects
print(mean_df)
```

```
   gender  math score  reading score  writing score
0  female   63.633205      72.608108      72.467181
1    male   68.728216      65.473029      63.311203
```

In [19]:

```
#in this command we will change a varible name so we can use it in the future
df.rename(columns = {'math score':'math_score'}, inplace = True)
```

In [20]:

```
# this code will give us the average number for the varible we placed
df['math_score'].mean()
```

Out[20]:

```
66.089
```

In [21]:

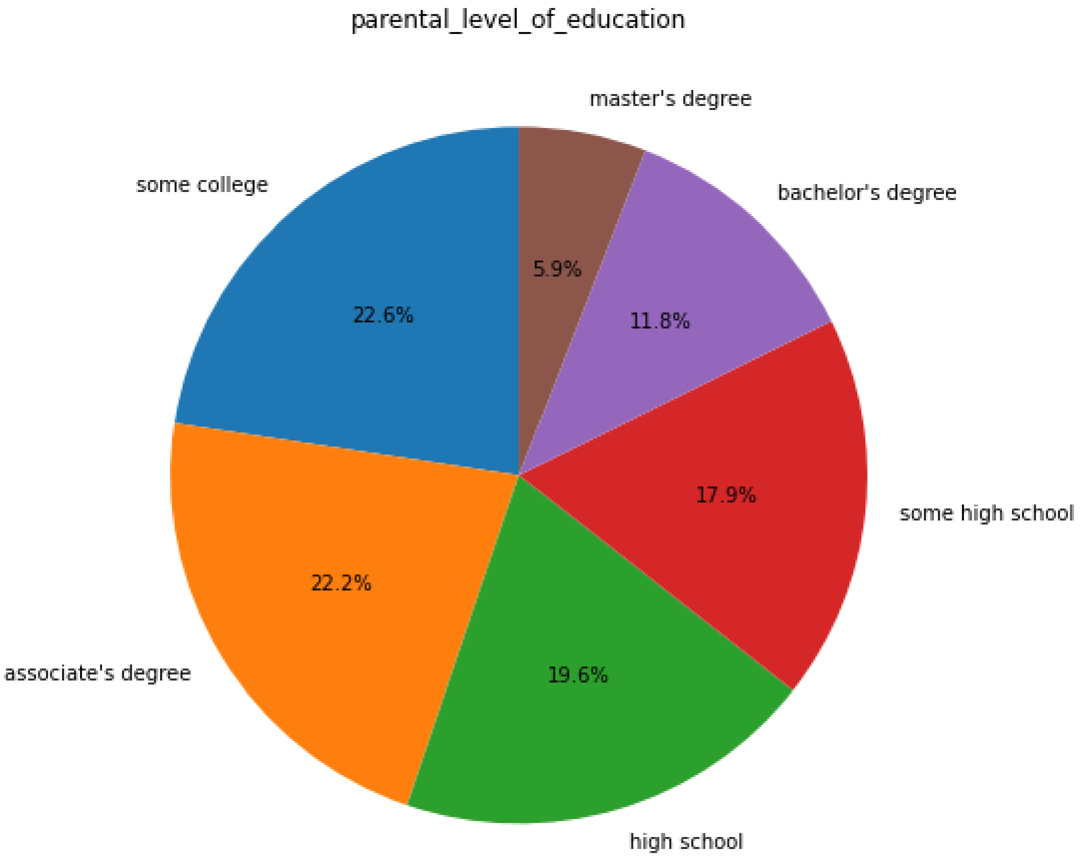
```
labels = df['parental_level_of_education'].value_counts().index
values = df['parental_level_of_education'].value_counts().values
```

In [22]:

```
# plt code will give us a chart from the ibrary that we imported earlier
#this chart describes the average number for the varible parental level of education
plt.figure(figsize=(8,8))
plt.pie(values, labels=labels, autopct='%1.1f%%', startangle = 90)
plt.title('parental_level_of_education')
plt.show
```

Out[22]:

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

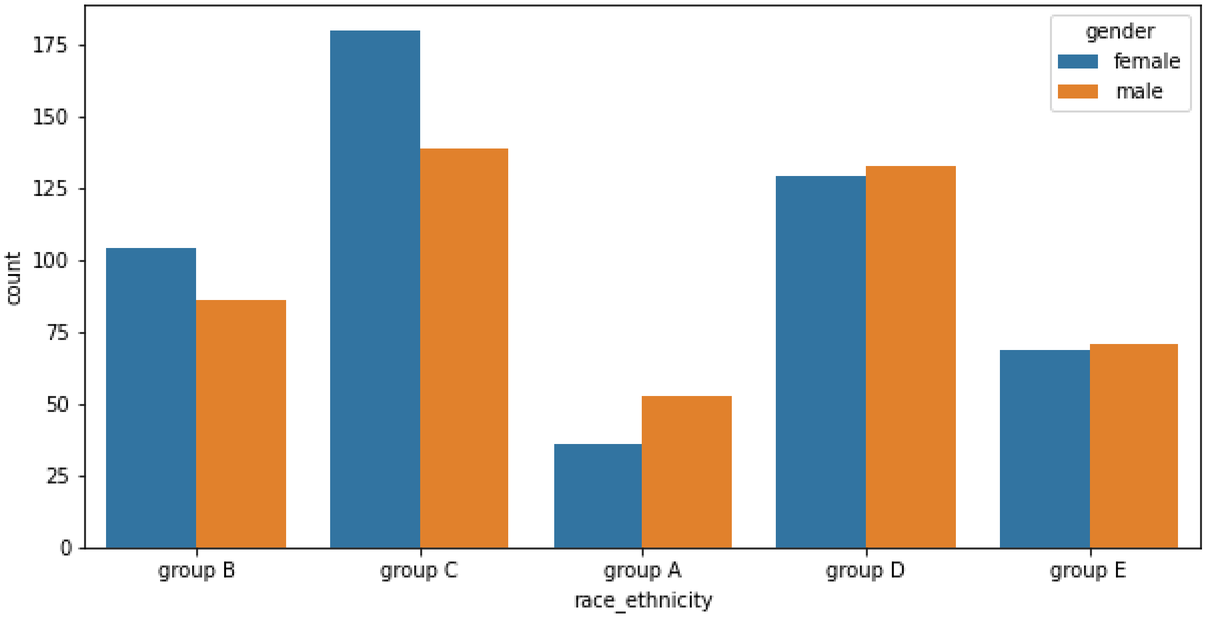


In [23]:

```
#in this command we will change a varible name so we can use it in the future
df.rename(columns = {'race/ethnicity':'race_ethnicity'}, inplace = True)
```

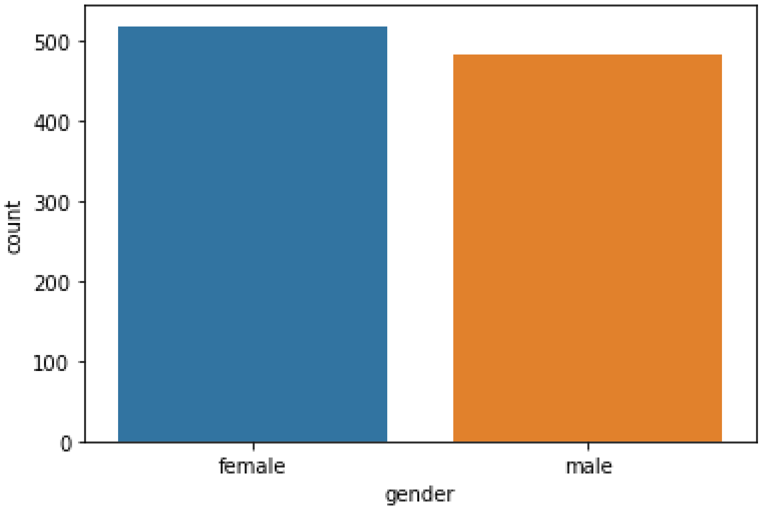
In [24]:

```
# this chart will show us The difference between the numbers of the varibles females
plt.figure(figsize=(10,5))
sb.countplot(x=df['race_ethnicity'],hue=df['gender']);
```



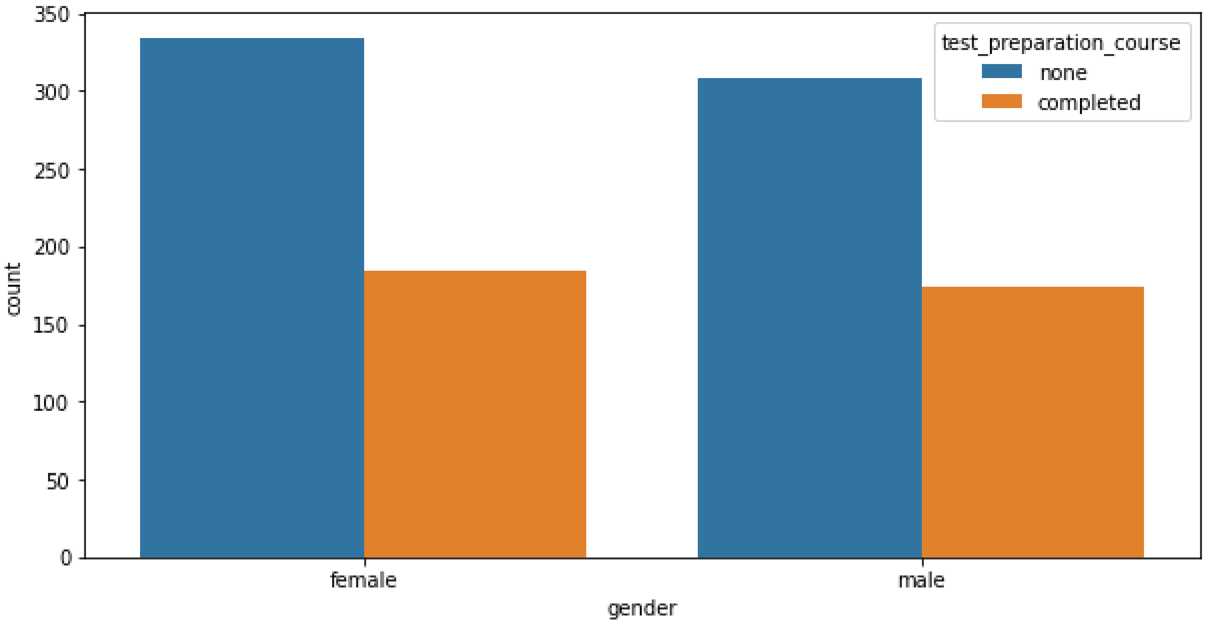
In [25]:

```
# chart for the difference between males and females numbers in the school
sb.countplot(data=df, x = 'gender');
```



In [26]:

```
# a chart showing us the students who have completed the test preparation course
plt.figure(figsize=(10,5))
sb.countplot(x=df['gender'],hue=df['test_preparation_course']);
```



In [27]:

```
#in this command we will change a varible name so we can use it in the future
df.rename(columns = {'reading score':'reading_score'}, inplace = True)
```

In [28]:

```
#this code will change a varible name so we can use it in the future
df.rename(columns = {'writing score':'writing_score'}, inplace = True)
```

In [29]:

```
group_df = df.groupby("race_ethnicity")
mean_df = group_df.mean()
```

In [30]:

```
mean_df = mean_df.reset_index()
```

In [31]:

```
# this code will print the average grade for the three subjects for each gorup
print(mean_df)
```

```
  race_ethnicity  math_score  reading_score  writing_score
0        group A   61.629213      64.674157      62.674157
1        group B   63.452632      67.352632      65.600000
2        group C   64.463950      69.103448      67.827586
3        group D   67.362595      70.030534      70.145038
4        group E   73.821429      73.028571      71.407143
```

In [32]:

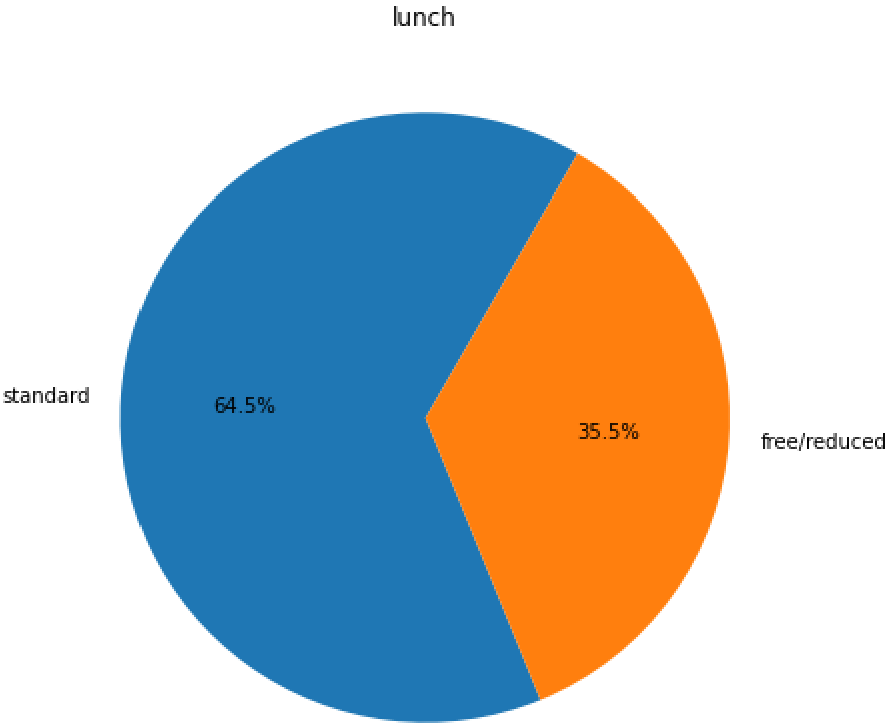
```
labels = df['lunch'].value_counts().index
values = df['lunch'].value_counts().values
```

In [33]:

```
# this chart will show us the most selected type of the lunch types and will print t
plt.figure(figsize=(7,7))
plt.pie(values, labels=labels, autopct='%1.1f%%', startangle = 60)
plt.title('lunch')
plt.show
```

Out[33]:

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



In [ ]:

```
```