```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib
from matplotlib import pyplot as plt
pd.options.mode.chained_assignment = None
```

In [2]:
```
sp=pd.read_csv('StudentsPerformance.csv')
sp.head(5)
```

Out[2]:

	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 [3]:
```
sp.describe()
```

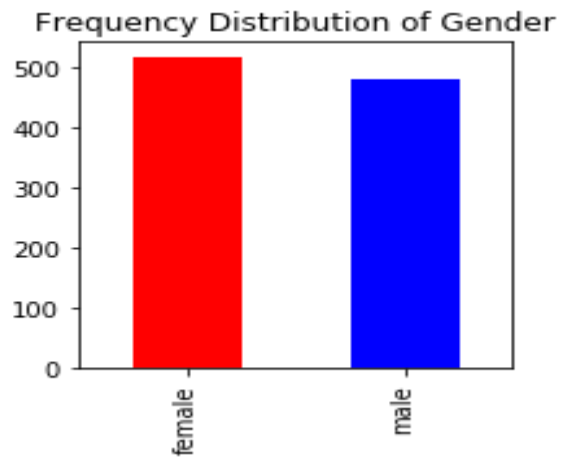
Out[3]:

	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 [4]: sp['gender'].value_counts().plot.bar(title='Frequency Distribution of Gender',color=['red','blue'],figsize=(3,3))

out[4]:

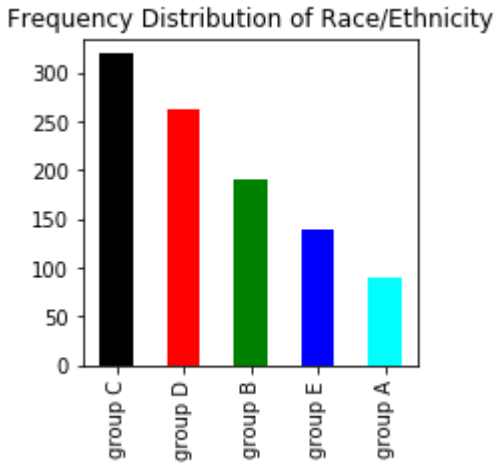
<matplotlib.axes._subplots.AxesSubplot at 0x13f16043ef0>



In [5]: sp['race/ethnicity'].value_counts().plot.bar(title='Frequency Distribution of Race/Ethnicity',color=['black', 'red', 'green', 'blue', 'cyan'],figsize=(3,3))

Out[5]:

<matplotlib.axes._subplots.AxesSubplot at 0x13f18110470>

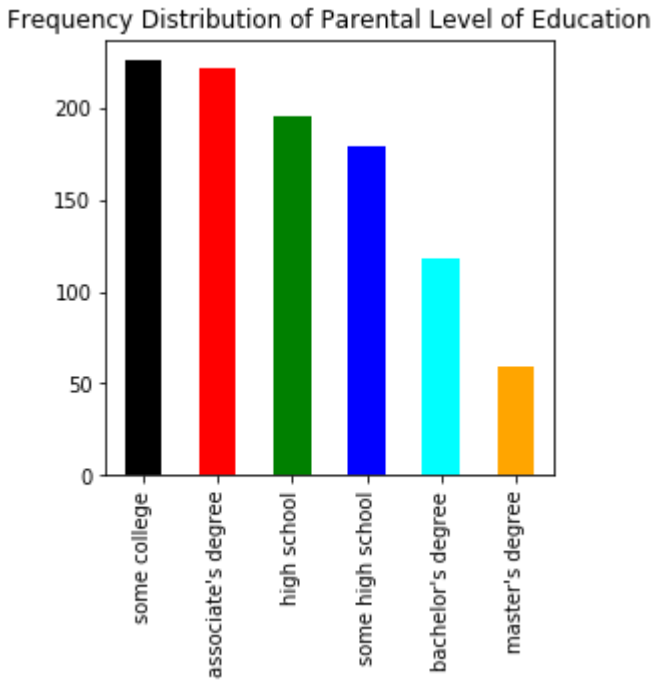


In [6]:

sp['parental level of education'].value_counts().plot.bar(title='Frequency Distribution of Parental Level of Education',color=['black', 'red', 'green', 'blue', 'cyan','orange'],figsize=(4,4))

Out[6]:

<matplotlib.axes._subplots.AxesSubplot at 0x13f18110908>

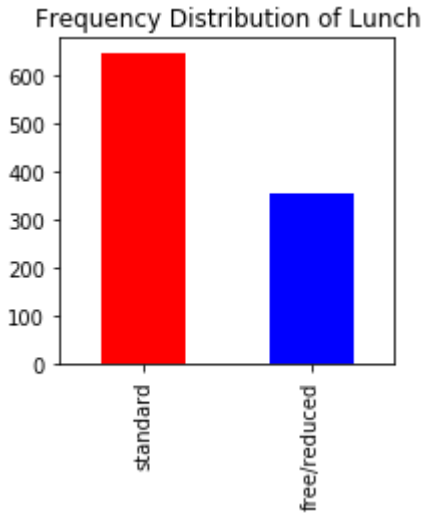


In [7]:

```
sp['lunch'].value_counts().plot.bar(title='Frequency Distribution of
Lunch',color=['red','blue'],figsize=(3,3))
```

Out[7]:

<matplotlib.axes._subplots.AxesSubplot at 0x13f181a2b70>

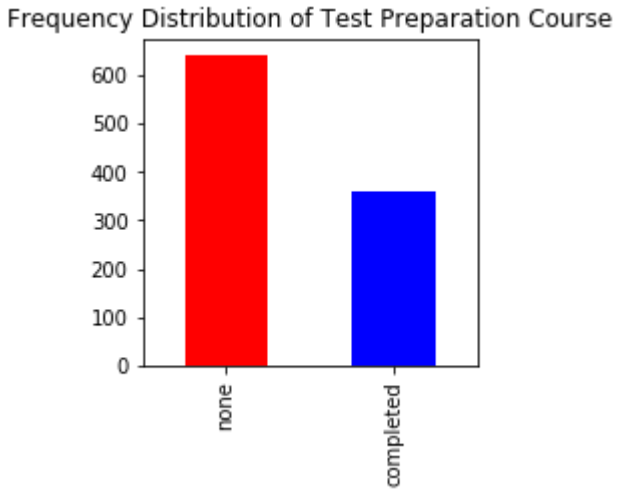


In [8]:

```
sp['test preparation course'].value_counts().plot.bar(title='Frequency Distribution of Test
Preparation Course',color=['red','blue'],figsize=(3,3))
```

Out[8]:

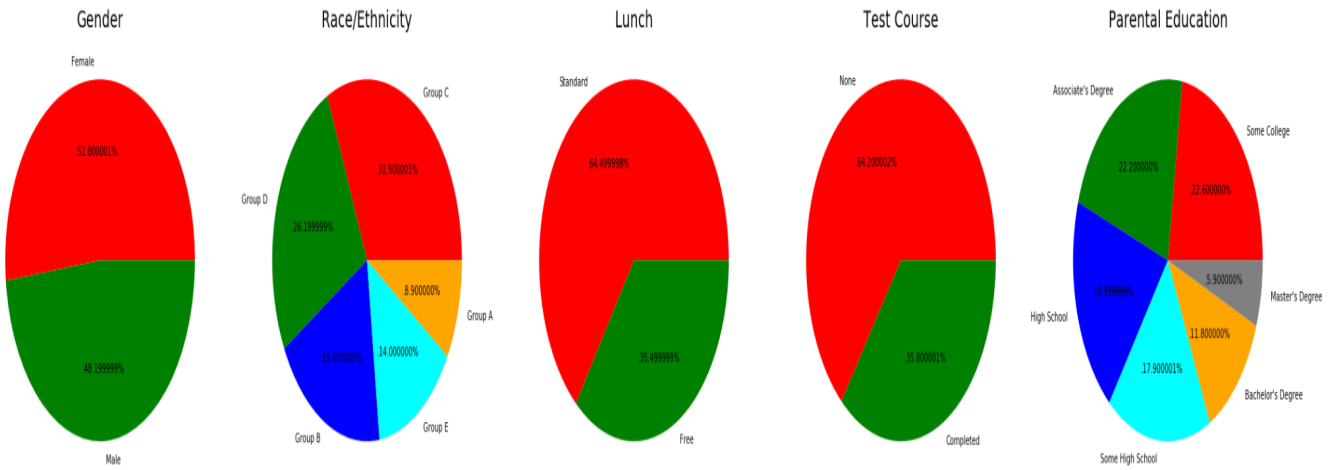
<matplotlib.axes._subplots.AxesSubplot at 0x13f1817fe48>



In [9]:

```
plt.rcParams['figure.figsize'] = (30, 12)

plt.subplot(1, 5, 1)
size = sp['gender'].value_counts()
labels = 'Female', 'Male'
color = ['red','green']


plt.pie(size, colors = color, labels = labels,autopct = '.%2f%%')
plt.title('Gender', fontsize = 20)
plt.axis('off')



plt.subplot(1, 5, 2)
size = sp['race/ethnicity'].value_counts()
labels = 'Group C', 'Group D','Group B','Group E','Group A'
color = ['red', 'green', 'blue', 'cyan','orange']

plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
plt.title('Race/Ethnicity', fontsize = 20)
plt.axis('off')



plt.subplot(1, 5, 3)
size = sp['lunch'].value_counts()
labels = 'Standard', 'Free'
color = ['red','green']

plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
```

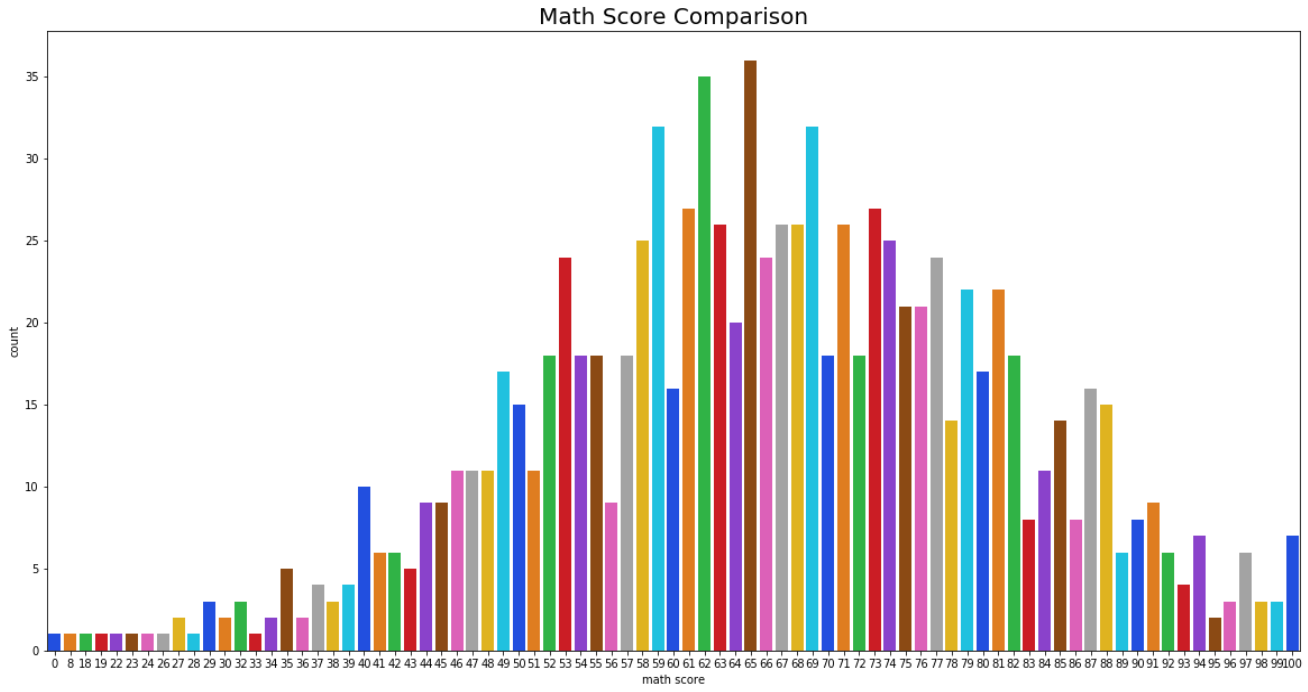
```
plt.title('Lunch', fontsize = 20)
plt.axis('off')


plt.subplot(1, 5, 4)
size = sp['test preparation course'].value_counts()
labels = 'None', 'Completed'
color = ['red','green']

plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
plt.title('Test Course', fontsize = 20)
plt.axis('off')


plt.subplot(1, 5, 5)
size = sp['parental level of education'].value_counts()
labels = 'Some College', "Associate's Degree",'High School','Some High School',"Bachelor's
Degree","Master's Degree"
color = ['red', 'green', 'blue', 'cyan','orange','grey']

plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
plt.title('Parental Education', fontsize = 20)
plt.axis('off')


plt.tight_layout()
plt.grid()

plt.show()
```



Number of Male and Female students is almost equal
Number students are greatest in Group C
Number of students who have standard lunch are greater
Number of students who have not enrolled in any test preparation course is greater
Number of students whose parental education is "Some College" is greater followed closely by "Associate's Degree"

In [10]:

```
plt.rcParams['figure.figsize'] = (20, 10)
sns.countplot(sp['math score'], palette = 'bright')
plt.title('Math Score Comparison',fontsize = 20)
plt.show()
```

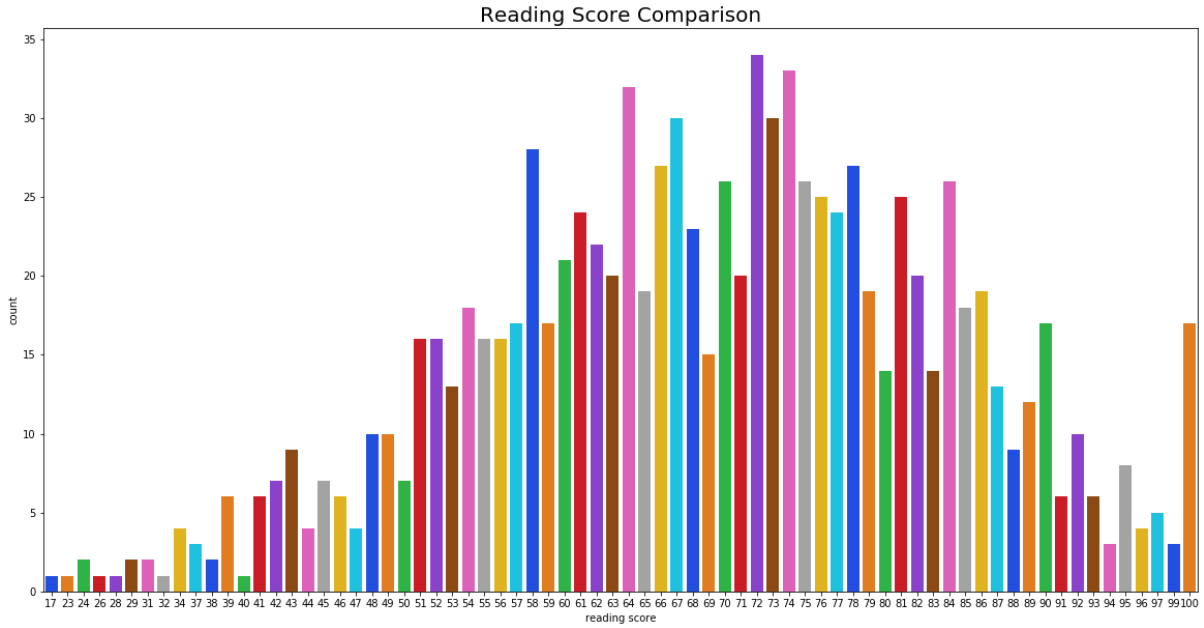


Maximum number of students had a Maths Score of 65

In [11]:

```
plt.rcParams['figure.figsize'] = (20, 10)
```

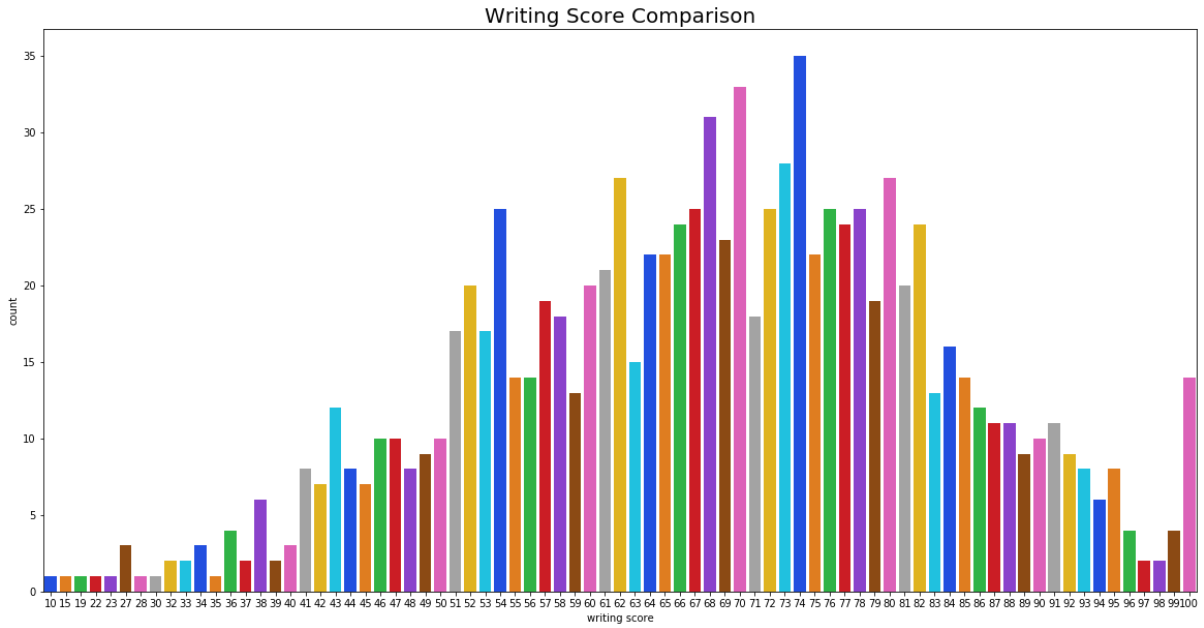
```
sns.countplot(sp['reading score'], palette = 'bright')
plt.title('Reading Score Comparison',fontsize = 20)
plt.show()
```



Maximum number of students had a Reading Score of 72

In [12]:

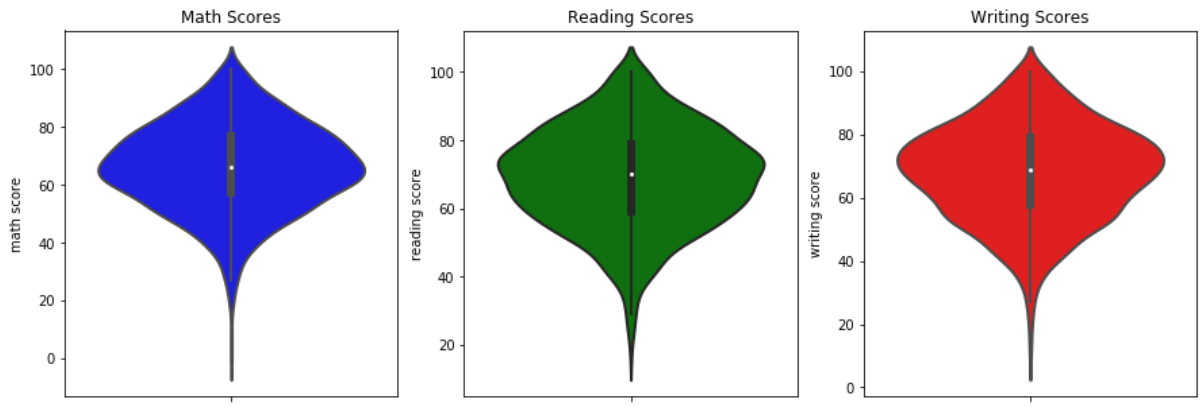
```
plt.rcParams['figure.figsize'] = (20, 10)
sns.countplot(sp['writing score'], palette = 'bright')
plt.title('Writing Score Comparison',fontsize = 20)
plt.show()
```



Maximum number of students had a Writing Score of 75

In [13]:

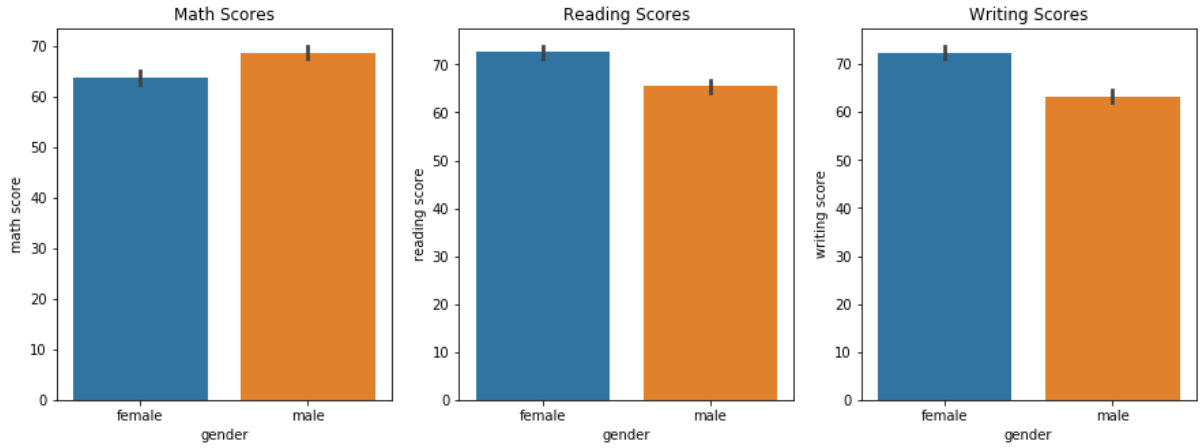
```
plt.figure(figsize=(15,5))
plt.subplot(131)
plt.title('Math Scores')
sns.violinplot(y='math score',data=sp,color='blue',linewidth=2)
plt.subplot(132)
plt.title('Reading Scores')
sns.violinplot(y='reading score',data=sp,color='green',linewidth=2)
plt.subplot(133)
plt.title('Writing Scores')
sns.violinplot(y='writing score',data=sp,color='red',linewidth=2)
plt.show()
```



Maximum number of students have scored between 60-80 marks in all 3 subjects

In [14]:

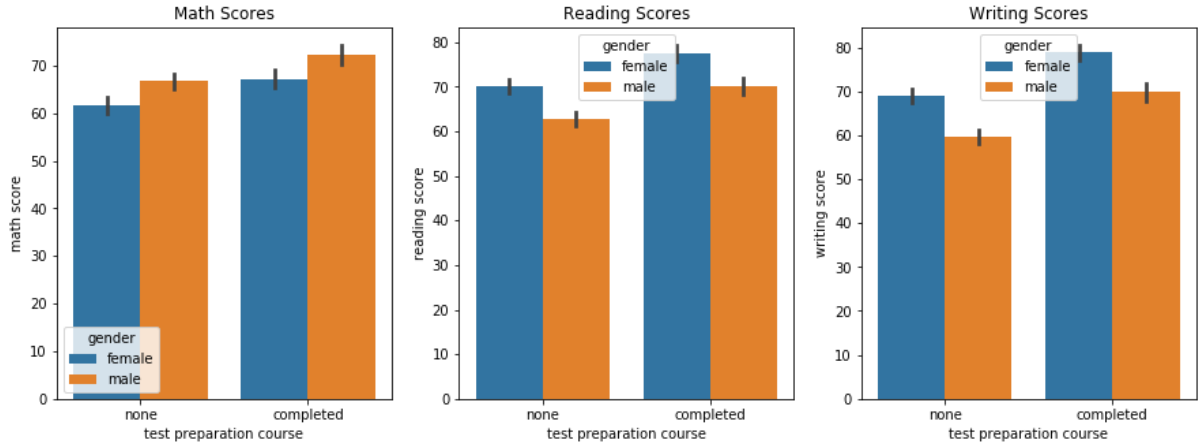
```
plt.figure(figsize=(15,5))
plt.subplot(131)
plt.title('Math Scores')
sns.barplot(x="gender", y="math score", data=sp)
plt.subplot(132)
plt.title('Reading Scores')
sns.barplot(x="gender", y="reading score", data=sp)
plt.subplot(133)
plt.title('Writing Scores')
sns.barplot(x="gender", y="writing score", data=sp)
plt.show()
```



Male students scored higher in Maths whereas Female students scored higher in Reading and Writing

In [15]:

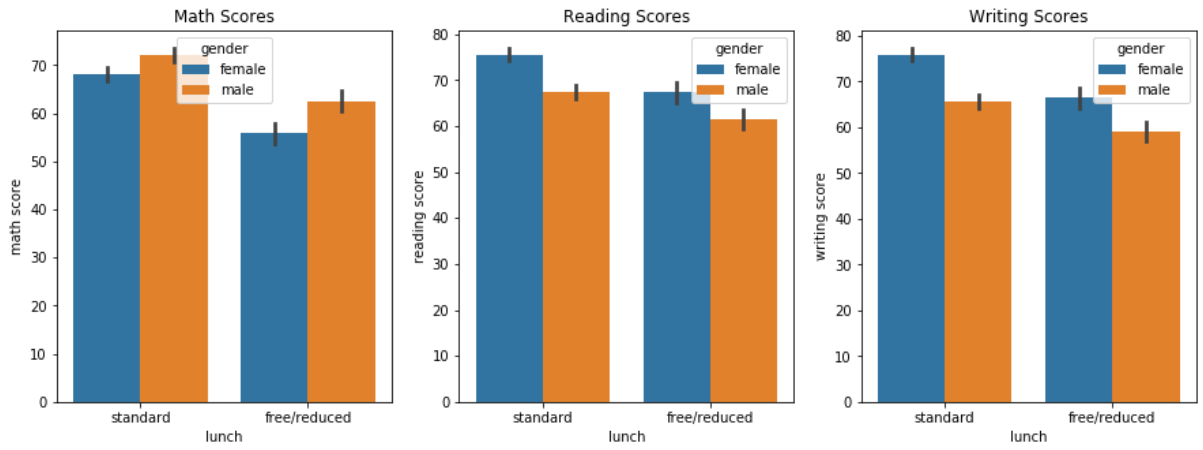
```
plt.figure(figsize=(15,5))
plt.subplot(131)
plt.title('Math Scores')
sns.barplot(hue="gender", y="math score", x="test preparation course", data=sp)
plt.subplot(132)
plt.title('Reading Scores')
sns.barplot(hue="gender", y="reading score", x="test preparation course", data=sp)
plt.subplot(133)
plt.title('Writing Scores')
sns.barplot(hue="gender", y="writing score", x="test preparation course", data=sp)
plt.show()
```



Students who completed the test preparation course acheived higher scored in all 3 subject

In [16]:

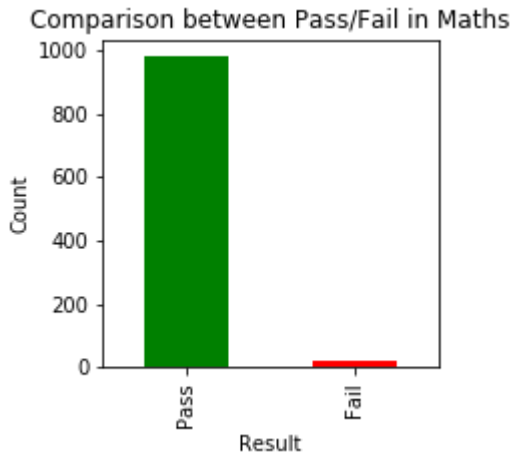
```
plt.figure(figsize=(15,5))
plt.subplot(131)
plt.title('Math Scores')
sns.barplot(hue="gender", y="math score", x="lunch", data=sp)
plt.subplot(132)
plt.title('Reading Scores')
sns.barplot(hue="gender", y="reading score", x="lunch", data=sp)
plt.subplot(133)
plt.title('Writing Scores')
sns.barplot(hue="gender", y="writing score", x="lunch", data=sp)
plt.show()
```



Students who chose standard lunch acheived higher scored in all 3 subject

In [17]:

```
passingmark=33
sp['pass_math'] = np.where(sp['math score']>= passingmark, 'Pass', 'Fail')
sp['pass_math'].value_counts(dropna = False).plot.bar(color=['green','red'], figsize = (3,3))

plt.title('Comparison between Pass/Fail in Maths')
plt.xlabel('Result')
plt.ylabel('Count')
plt.show()
sp['pass_math'].value_counts()
```

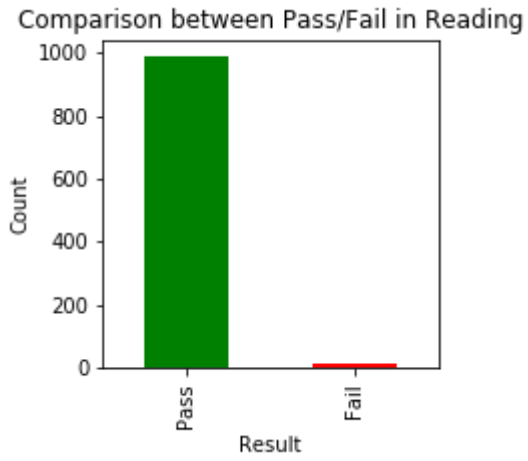


Out[17]:
Pass 981
Fail 19

Name: pass_math, dtype: int64

In [18]:

```
sp['pass_read'] = np.where(sp['reading score']>= passingmark, 'Pass', 'Fail')
sp['pass_read'].value_counts(dropna = False).plot.bar(color = ['green','red'], figsize = (3,3))
plt.title('Comparison between Pass/Fail in Reading')
plt.xlabel('Result')
plt.ylabel('Count')
plt.show()
sp['pass_read'].value_counts()
```

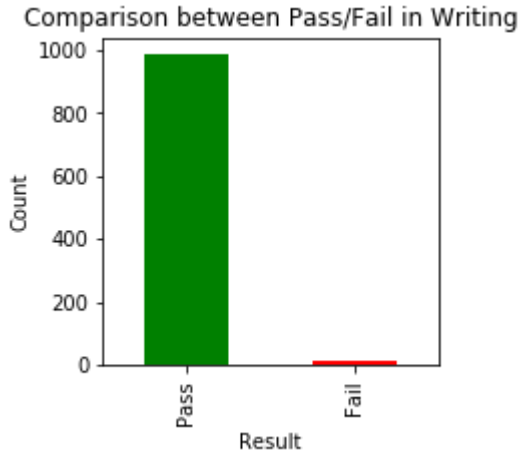


Out[18]:
Pass 989
Fail 11

Name: pass_read, dtype: int64

In [19]:

```
sp['pass_write'] = np.where(sp['writing score']>= passingmark, 'Pass', 'Fail')
sp['pass_write'].value_counts(dropna = False).plot.bar(color = ['green','red'], figsize = (3,3))
plt.title('Comparison between Pass/Fail in Writing')
plt.xlabel('Result')
plt.ylabel('Count')
plt.show()
sp['pass_write'].value_counts()
```

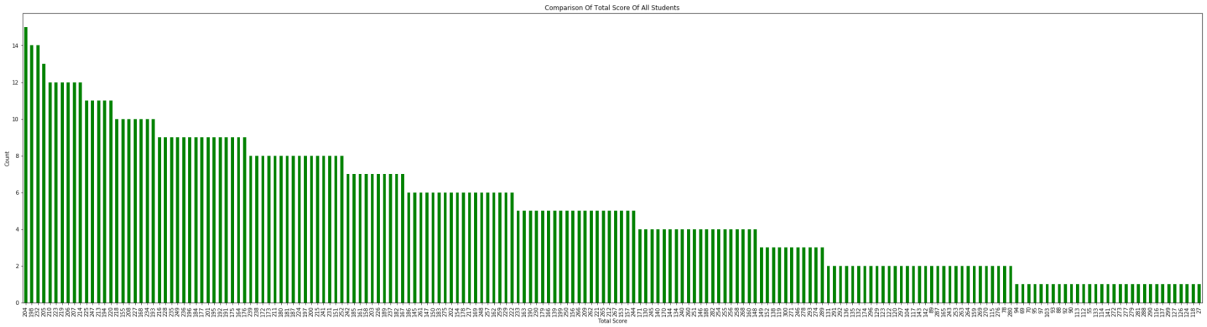


Out[19]:

```
Pass    988
Fail     12
Name: pass_write, dtype: int64
```

In [20]:

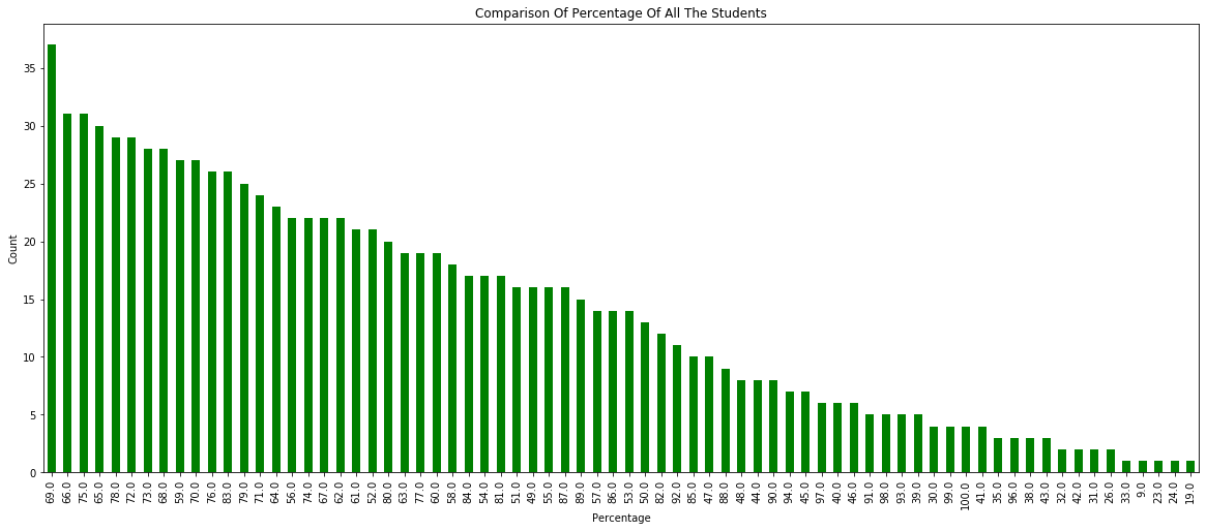
```
sp['total_score'] = sp['math score'] + sp['reading score'] + sp['writing score']

sp['total_score'].value_counts(normalize = True)
sp['total_score'].value_counts(dropna = True).plot.bar(color = 'green', figsize = (40,10))

plt.title('Comparison Of Total Score Of All Students')
plt.xlabel('Total Score')
plt.ylabel('Count')
plt.show()
```



Maximum number of students had a Total Score of 204

In [21]:

```
#For calculating percentage scored by each student
from math import *
sp['percentage'] = sp['total_score']/3
for i in range(0, 1000):
  sp['percentage'][i] = ceil(sp['percentage'][i])

sp['percentage'].value_counts(normalize = True)
sp['percentage'].value_counts(dropna = False).plot.bar(figsize = (20, 8), color = 'green')

plt.title('Comparison Of Percentage Of All The Students')
plt.xlabel('Percentage')
plt.ylabel('Count')
plt.show()
```

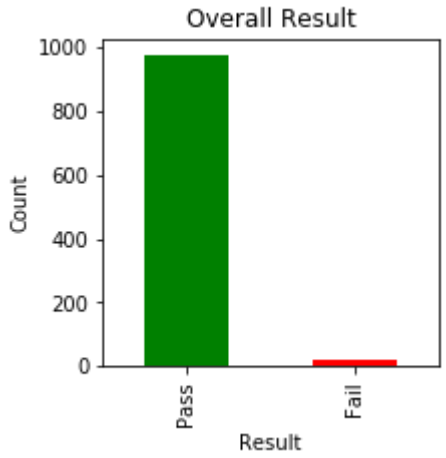


Maximum number of students scored 69%

In [22]:

```
#Assigning Pass/Fail through overall score

sp['result'] = sp.apply(lambda x : 'Fail' if x['pass_math'] == 'Fail' or
                        x['pass_read'] == 'Fail' or x['pass_write'] == 'Fail'
                        else 'Pass', axis = 1)
```

```
sp['result'].value_counts(dropna = False).plot.bar(color = ['green','red'], figsize = (3, 3))
plt.title('Overall Result')
plt.xlabel('Result')
plt.ylabel('Count')
plt.show()

sp['result'].value_counts()
```



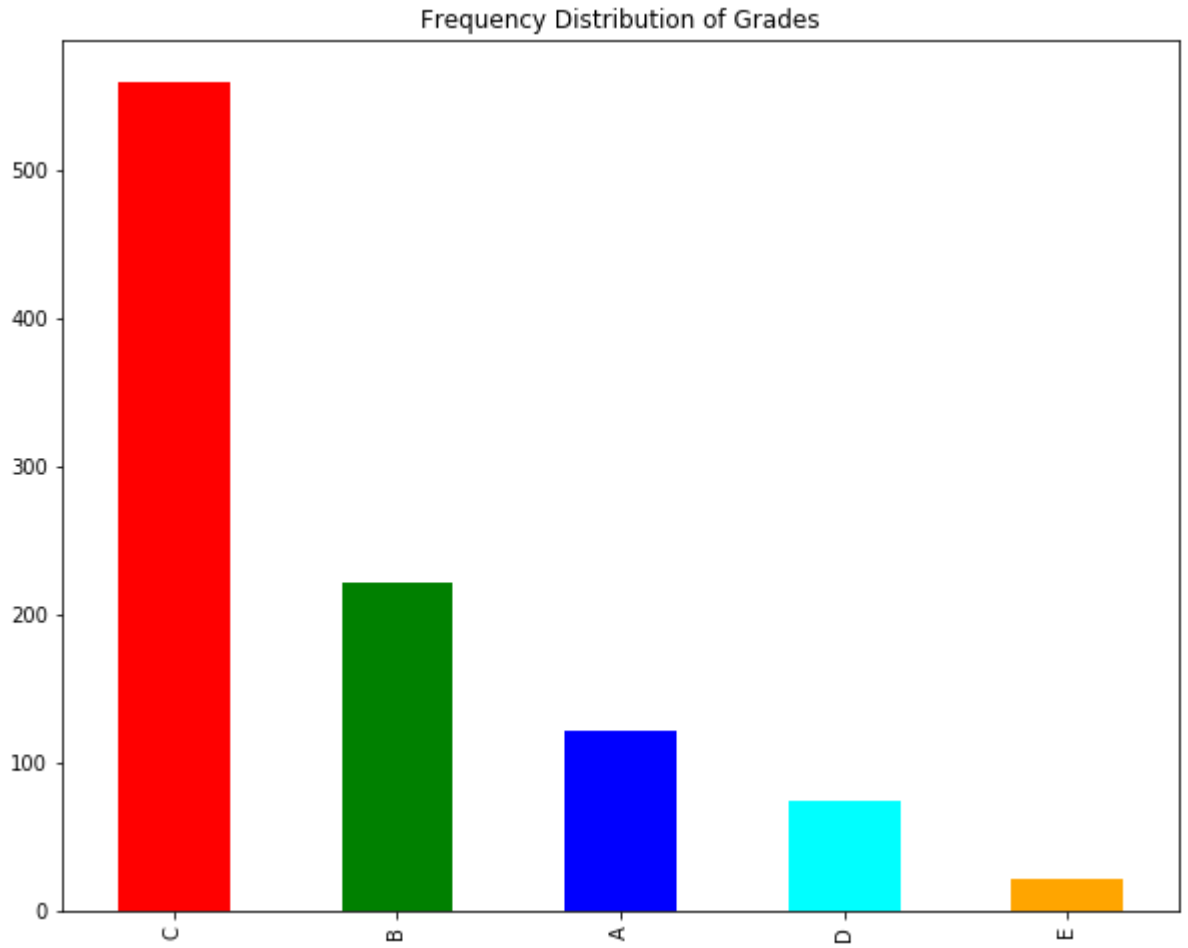
Out[22]:
Pass 978
Fail 22
Name: result, dtype: int64
In [23]:

```
# Assigning grades according to marks scored by students
# 0  - 33 marks : grade E
# 33 - 50 marks : grade D
# 50 - 75 marks : grade C
# 75 - 90 marks : grade B
# 90 - 100 marks : grade A

def calcgrade(percentage,result):

  if result == 'Fail':
    return 'E'
  if(percentage >= 85):
    return 'A'
  if(percentage >= 75):
    return 'B'
  if(percentage >= 50):
    return 'C'
  if(percentage >=33):
    return 'D'
  else:
    return 'E'

sp['grade']= sp.apply(lambda x: calcgrade(x['percentage'], x['result']), axis = 1 )
```

```
sp['grade'].value_counts().plot.bar(title='Frequency Distribution of Grades',color=['red', 'green', 'blue', 'cyan','orange'],figsize=(10,8))
```

Out[23]:

<matplotlib.axes._subplots.AxesSubplot at 0x13f1871de10>



In [24]:

```
sp['grade'].value_counts()
```

Out[24]:

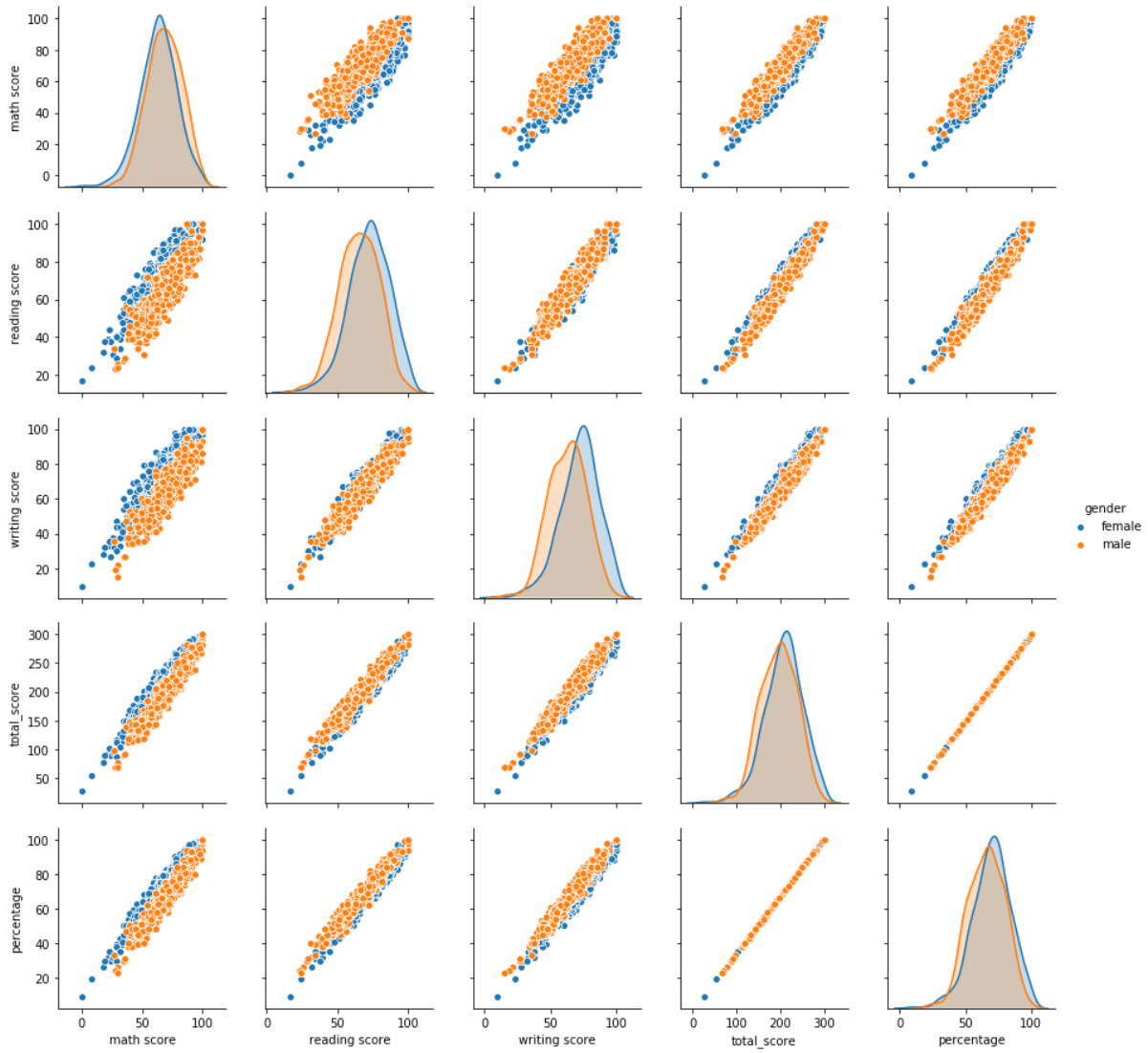
```
C    560
B    222
A    122
D     74
E     22
```

Name: grade, dtype: int64

Maximum number of students received Grade C

In [25]:

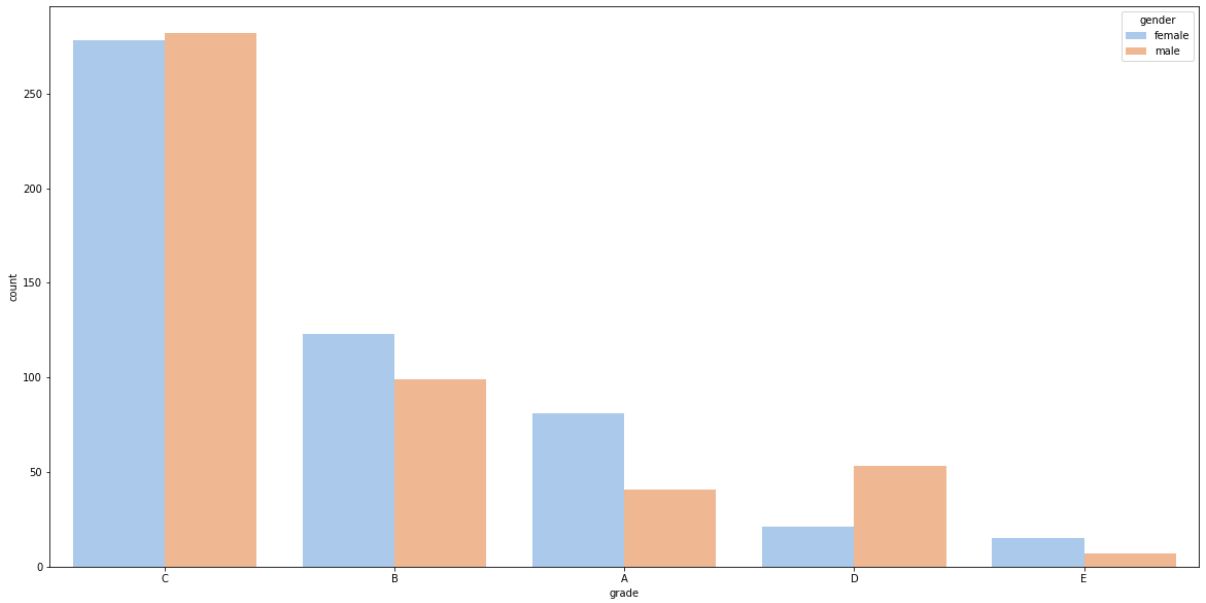
```
sns.pairplot(sp,hue = 'gender')
plt.show()
```



In [26]:

#Comparison between gender and grade acquired by student

```
sns.countplot(x = sp['grade'], data=sp, hue = sp['gender'], palette = 'pastel')
plt.show()
```



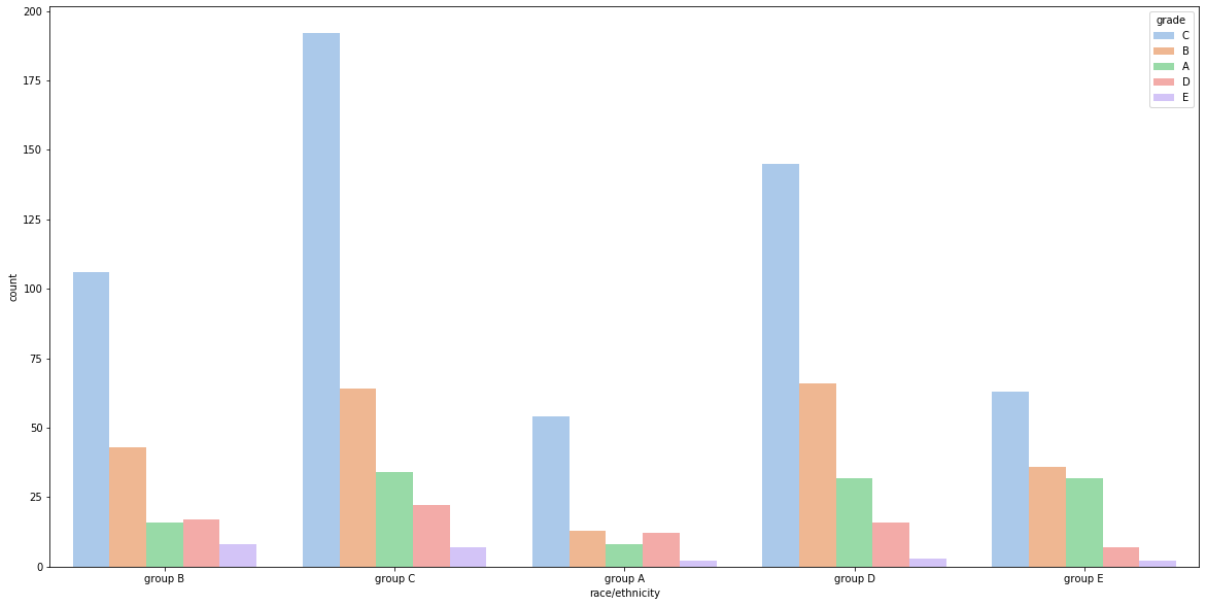
Maximum number of students that acheived Grade A and Grade B were Females

In [27]:

#Comparison between Race/Ethnicity and Grade acquired by student

sns.countplot(x = sp['race/ethnicity'], data=sp, hue = sp['grade'], palette = 'pastel')

plt.show()



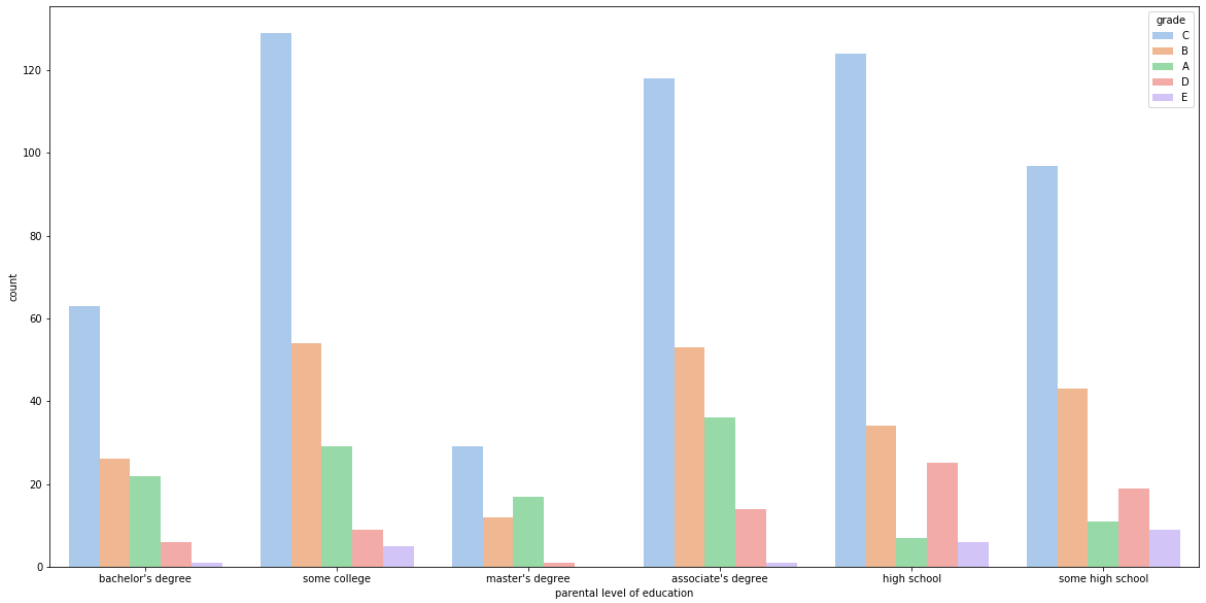
Maximum number of students that acheived high grades like Grade A and Grade B were from Group C and Group D

In [28]:

#Comparison between parental level of education and grade acquired by student

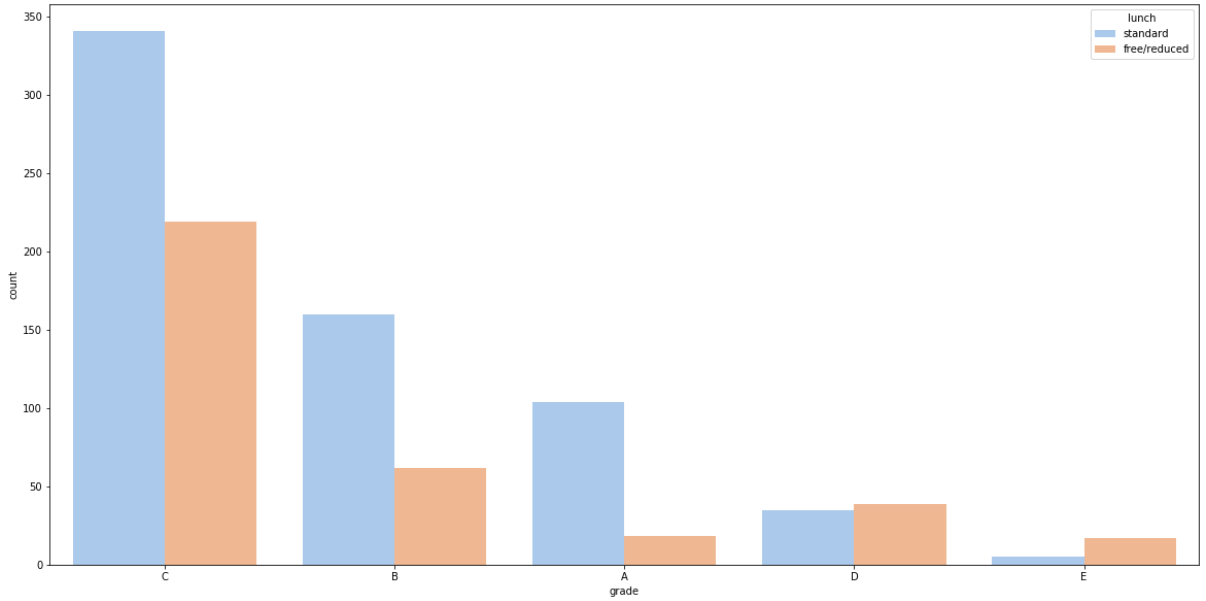
sns.countplot(x = sp['parental level of education'], data=sp, hue = sp['grade'], palette = 'pastel')

plt.show()



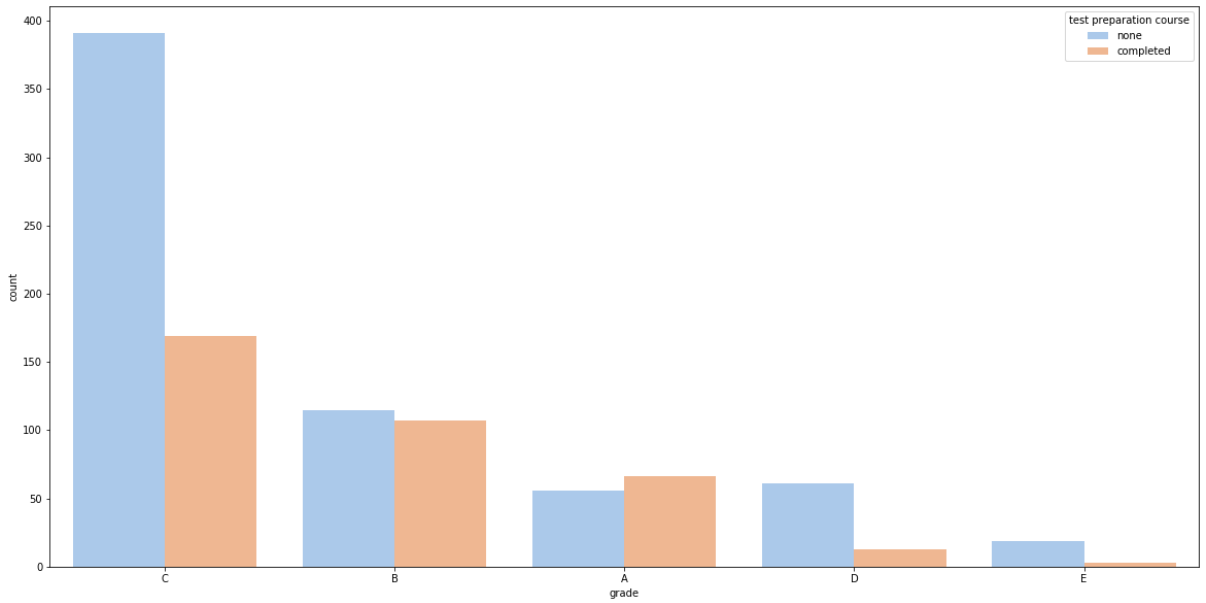
Number of students whose parents had an associate's degree scored higher than other students

In [29]:
#Comparison between lunch and grade acquired by student
sns.countplot(x = sp['grade'], data=sp, hue = sp['lunch'], palette = 'pastel')
plt.show()



Number of students who chose standard lunch scored higher than other students

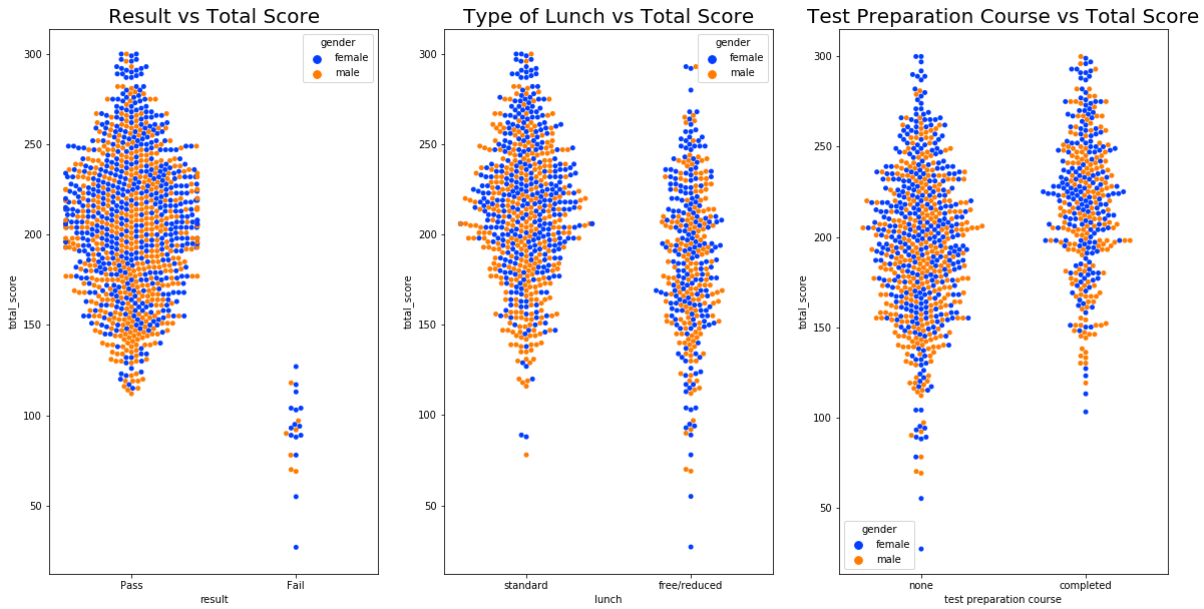
In [30]:
#Comparison between test prep course and grade acquired by student
sns.countplot(x = sp['grade'], data=sp, hue = sp['test preparation course'], palette = 'pastel')
plt.show()



Number of students that completed the test preparation course and got Grade A were higher

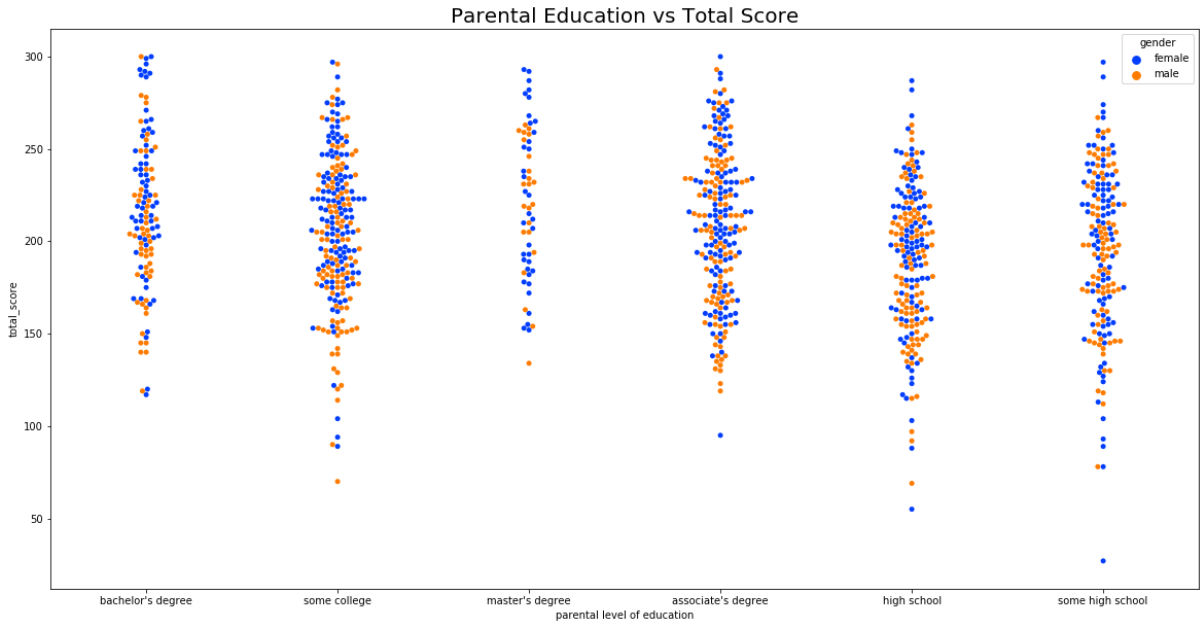
In [35]:

```
plt.rcParams['figure.figsize'] = (20,10)

plt.subplot(1, 3, 1)
sns.swarmplot(sp['result'], sp['total_score'], hue = sp['gender'], palette = 'bright')
plt.title('Result vs Total Score', fontsize = 20)

plt.subplot(1, 3, 2)
sns.swarmplot(sp['lunch'], sp['total_score'], hue = sp['gender'], palette = 'bright')
plt.title('Type of Lunch vs Total Score', fontsize = 20)

plt.subplot(1, 3, 3)
sns.swarmplot(sp['test preparation course'], sp['total_score'], hue = sp['gender'], palette = 'bright')
plt.title('Test Preparation Course vs Total Score', fontsize = 20)
plt.show()

plt.show()
```



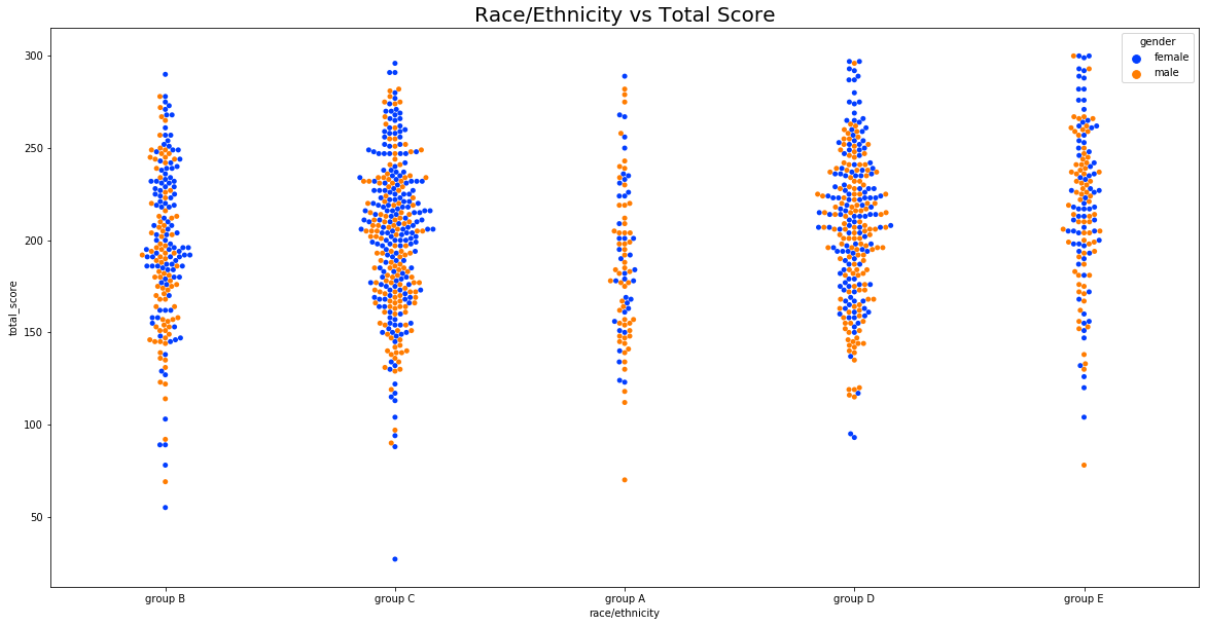
In [32]:

```
plt.rcParams['figure.figsize'] = (20,10)
plt.subplot(1, 1, 1)
sns.swarmplot(sp['parental level of education'], sp['total_score'], hue = sp['gender'], palette = 'bright')
plt.title('Parental Education vs Total Score', fontsize = 20)
plt.show()
```



In [33]:

```
plt.rcParams['figure.figsize'] = (20,10)
plt.subplot(1, 1, 1)
sns.swarmplot(sp['race/ethnicity'], sp['total_score'], hue = sp['gender'], palette = 'bright')
plt.title('Race/Ethnicity vs Total Score', fontsize = 20)
plt.show()
```



In [34]:

```
#Visualizing realtions between various attributes using heatmap
plt.figure(figsize=(20,10))
plt.rcParams['figure.figsize'] = (18, 16)
sp_corr = sp.corr()
ax = sns.heatmap(sp_corr, annot=True,cmap="Greens")
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)
sp_corr
```

Out[34]:

	math score	reading score	writing score	total_score	percentage
math score	1.000000	0.817580	0.802642	0.918746	0.918521
reading score	0.817580	1.000000	0.954598	0.970331	0.970271
writing score	0.802642	0.954598	1.000000	0.965667	0.965422
total_score	0.918746	0.970331	0.965667	1.000000	0.999813

percentage	0.918521	0.970271	0.965422	0.999813	1.000000