In [1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:

```python
df=pd.read_csv("/home/mca01/Downloads/StudentsPerformance.csv")
```

In [4]:

```python
df.head(15)
```

Out[4]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	readin scor
0	female	group B	bachelor's degree	standard	none	72	7
1	female	group C	some college	standard	completed	69	9
2	female	group B	master's degree	standard	none	90	9
3	male	group A	associate's degree	free/reduced	none	47	5
4	male	group C	some college	standard	none	76	7
5	female	group B	associate's degree	standard	none	71	8
6	female	group B	some college	standard	completed	88	9
7	male	group B	some college	free/reduced	none	40	4
8	male	group D	high school	free/reduced	completed	64	6
9	female	group B	high school	free/reduced	none	38	6
10	male	group C	associate's degree	standard	none	58	5
11	male	group D	associate's degree	standard	none	40	5
12	female	group B	high school	standard	none	65	8
13	male	group A	some college	standard	completed	78	7
14	female	group A	master's degree	standard	none	50	5

In [5]:

```python
df.shape
```

Out[5]:
```
(1000, 8)
```

In [6]:
```
df.dtypes.value_counts()
```

Out[6]:
```
object    5
int64     3
dtype: int64
```

In [7]:
```
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 [8]:
```
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

Handle the Missing Value

In [9]:
```
df.isnull().sum()
```

Loading [MathJax]/extensions/Safe.js

Out[9]:

```
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
```

Making list of columns having missing value

In [10]:

```
data = df
coln=[]
miss=[]
coln.extend(data.columns)
for i in coln:
   t=data[i].isnull

   if t!=0:
     miss.append(i)
   else:
     continue
print(miss)
```

```
['gender', 'race/ethnicity', 'parental level of education', 'lunch', 'test p
reparation course', 'math score', 'reading score', 'writing score']
```

In [11]:

```
pd.options.mode.chained_assignment=None
for j in miss:
  q=data[j].dtypes
  if(q=='int64' or q=='float64'):
    f=data[j]
    for k in range(data.shape[0]):
        if(f[k]<0 or f[k]>100):
            f[k]=(np.nan)
  else:
      data.fillna(method='bfill')

data.head(20)
```

Loading [MathJax]/extensions/Safe.js

Out[11]:

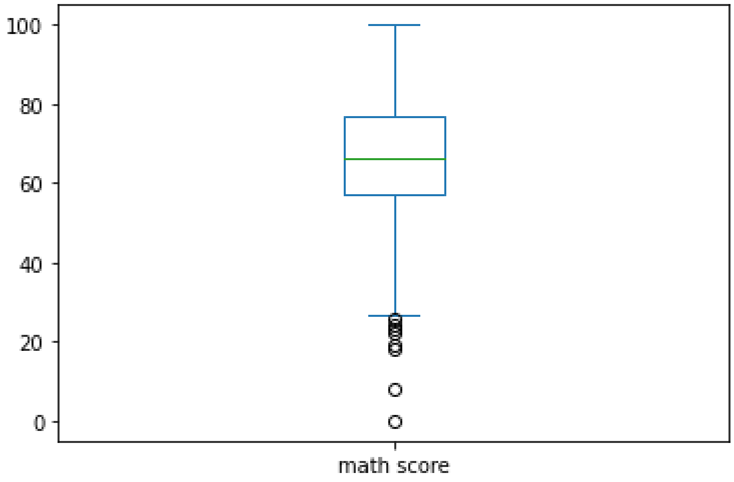
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	readin scor
0	female	group B	bachelor's degree	standard	none	72	7
1	female	group C	some college	standard	completed	69	9
2	female	group B	master's degree	standard	none	90	9
3	male	group A	associate's degree	free/reduced	none	47	5
4	male	group C	some college	standard	none	76	7
5	female	group B	associate's degree	standard	none	71	8
6	female	group B	some college	standard	completed	88	9
7	male	group B	some college	free/reduced	none	40	4
8	male	group D	high school	free/reduced	completed	64	6
9	female	group B	high school	free/reduced	none	38	6
10	male	group C	associate's degree	standard	none	58	5
11	male	group D	associate's degree	standard	none	40	5
12	female	group B	high school	standard	none	65	8
13	male	group A	some college	standard	completed	78	7
14	female	group A	master's degree	standard	none	50	5
15	female	group C	some high school	standard	none	69	7
16	male	group C	high school	standard	none	88	8
17	female	group B	some high school	free/reduced	none	18	3
18	male	group C	master's degree	free/reduced	completed	46	4
19	female	group C	associate's degree	free/reduced	none	54	5

Loading [MathJax]/extensions/Safe.js

Review of Parents

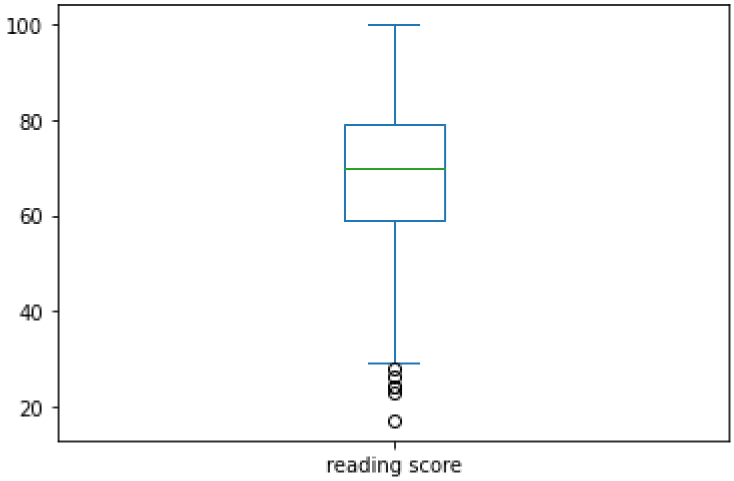
```
In [12]: data['math score'].plot(kind='box')
```

```
Out[12]: <AxesSubplot:>
```



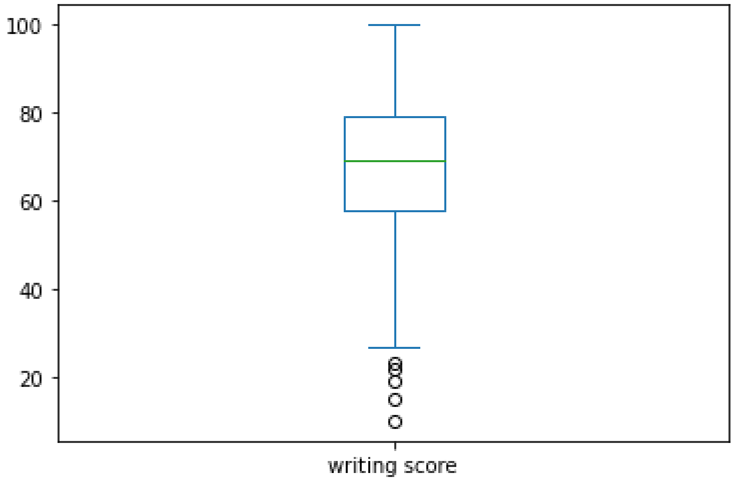
```
In [13]: data['reading score'].plot(kind='box')
```

```
Out[13]: <AxesSubplot:>
```



```
In [14]: data['writing score'].plot(kind='box')
```

```
Out[14]: <AxesSubplot:>
```



In [15]:
```
data.head()
```

Out[15]:

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

Outliers Removel

In [16]:
```
q1=data['math score'].quantile(0.25)
q3=data['math score'].quantile(0.75)
iqr = q3-q1

lowerlimit=q1 - 1.5*iqr
upperlimit =q3 + 1.5*iqr

print("Q1",q1, "\nQ3:" , q3,"\nIQR:",iqr, "\nLOWER LIMIT",lowerlimit,"\nUPPE
```

```
Q1 57.0
Q3: 77.0
IQR: 20.0
LOWER LIMIT 27.0
UPPER LIMIT 107.0
```

In [17]:
```
data[(data['math score']<lowerlimit)|(data['math score']>upperlimit)]
```

Loading [MathJax]/extensions/Safe.js

Out[17]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	readi sco
17	female	group B	some high school	free/reduced	none	18	
59	female	group C	some high school	free/reduced	none	0	
145	female	group C	some college	free/reduced	none	22	
338	female	group B	some high school	free/reduced	none	24	
466	female	group D	associate's degree	free/reduced	none	26	
787	female	group B	some college	standard	none	19	
842	female	group B	high school	free/reduced	completed	23	
980	female	group B	high school	free/reduced	none	8	

In [18]:

```
data[(data['math score']<lowerlimit)&(data['math score']>upperlimit)]
```

Out[18]:

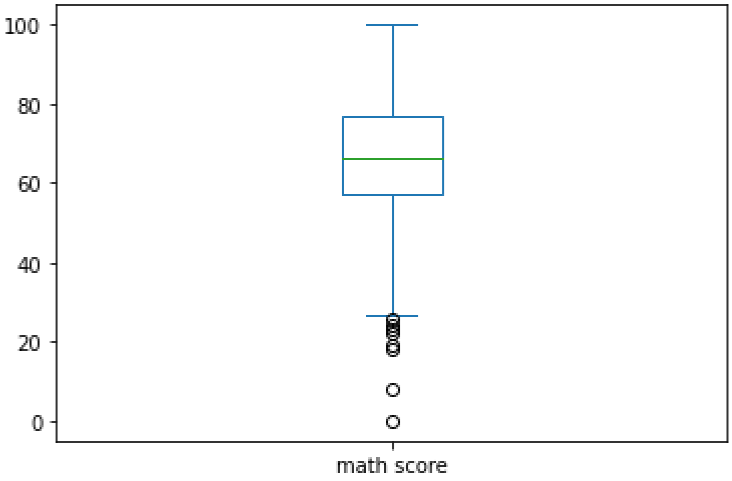
gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writir scoı

In [19]:

```
data['math score'].plot(kind='box')
```

Out[19]:

```
<AxesSubplot:>
```



Zscore Scaling

In [20]:
```
data
```

Out[20]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	readi sco
0	female	group B	bachelor's degree	standard	none	72	
1	female	group C	some college	standard	completed	69	
2	female	group B	master's degree	standard	none	90	
3	male	group A	associate's degree	free/reduced	none	47	
4	male	group C	some college	standard	none	76	
...	...	...	...	...	...	...	
995	female	group E	master's degree	standard	completed	88	
996	male	group C	high school	free/reduced	none	62	
997	female	group C	high school	free/reduced	completed	59	
998	female	group D	some college	standard	completed	68	
999	female	group D	some college	free/reduced	none	77	

1000 rows × 8 columns

In [21]:
```
new_data=data
from scipy import stats
```

In [22]:
```
columns=['math score','reading score','writing score']
new_data[columns]= stats.zscore(new_data[columns])
new_data
```

Out[22]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	
0	female	group B	bachelor's degree	standard	none	0.390024	0
1	female	group C	some college	standard	completed	0.192076	1
2	female	group B	master's degree	standard	none	1.577711	1
3	male	group A	associate's degree	free/reduced	none	-1.259543	-0
4	male	group C	some college	standard	none	0.653954	0
...	...	...	...	...	...	...	
995	female	group E	master's degree	standard	completed	1.445746	2
996	male	group C	high school	free/reduced	none	-0.269803	-0
997	female	group C	high school	free/reduced	completed	-0.467751	0
998	female	group D	some college	standard	completed	0.126093	0
999	female	group D	some college	free/reduced	none	0.719937	1

1000 rows × 8 columns

MinMax scaling

In [23]:

```
new_data1=data
```

In [30]:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
```

In [31]:

```
col=['math score','reading score','writing score']
scaler.fit(new_data1[col])
new_data1[col]=scaler.transform(new_data1[col])
```

In [32]:

```
new_data1
```

Out[32]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	read sc
0	female	group B	bachelor's degree	standard	none	0.72	0.662
1	female	group C	some college	standard	completed	0.69	0.879
2	female	group B	master's degree	standard	none	0.90	0.939
3	male	group A	associate's degree	free/reduced	none	0.47	0.481
4	male	group C	some college	standard	none	0.76	0.734
...	...	...	...	...	...	...	
995	female	group E	master's degree	standard	completed	0.88	0.987
996	male	group C	high school	free/reduced	none	0.62	0.457
997	female	group C	high school	free/reduced	completed	0.59	0.650
998	female	group D	some college	standard	completed	0.68	0.734
999	female	group D	some college	free/reduced	none	0.77	0.831

1000 rows × 8 columns

In [ ]:

Loading [MathJax]/extensions/Safe.js