In [8]:

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

In [9]:

```
df=pd.read_csv('health care diabetes.csv')
df.head()
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunc
0	6	148	72	35	0	33.6	0.
1	1	85	66	29	0	26.6	0.
2	8	183	64	0	0	23.3	0.
3	1	89	66	23	94	28.1	0.
4	0	137	40	35	168	43.1	2.

Descriptive Analysis

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Pregnancies                 768 non-null int64
Glucose                     768 non-null int64
BloodPressure               768 non-null int64
SkinThickness               768 non-null int64
Insulin                     768 non-null int64
BMI                         768 non-null float64
DiabetesPedigreeFunction    768 non-null float64
Age                         768 non-null int64
Outcome                     768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [11]:

```
df.describe()
```

Out[11]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diab
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

Insights from Descriptive Analysis

There is 768 observations of 9 variable. Independent variables are Pregnencies , Glucose, BloodPressure, Insulin, BMI and DiabetesPedigree Function. Age is Outcome Variable. Average Age of Patients are 33.24 with minimum being 21 and maximum 81. Avg. value of independent variables are Preg = 3.845052,Glucose = 120.894531, BP = 69.105469, ST=20.536458, Insulin = 79.799479, BMI = 31.992578 DPF = 0.471876 . Variation in variables can be easily observed from table below :->

In [12]:

```
print("Standard Deviation of each variables are ==> ")
df.apply(np.std)
```

```
Standard Deviation of each variables are ==>
```

Out[12]:

```
Pregnancies                 3.367384
Glucose                    31.951796
BloodPressure              19.343202
SkinThickness              15.941829
Insulin                   115.168949
BMI                         7.879026
DiabetesPedigreeFunction    0.331113
Age                        11.752573
Outcome                     0.476641
dtype: float64
```

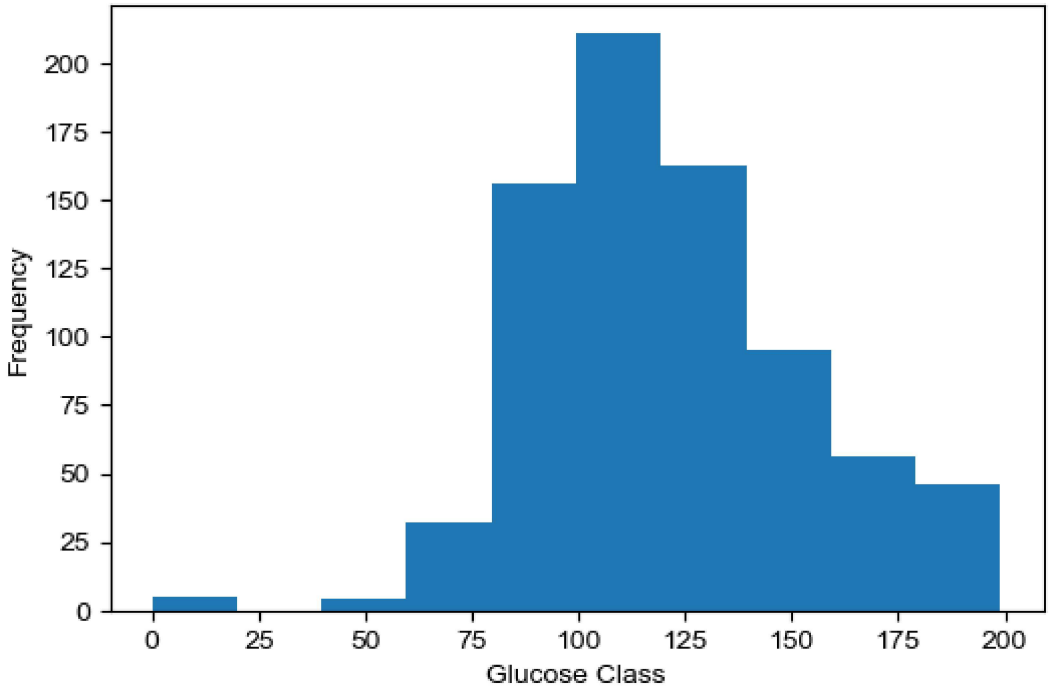
Treating Missing Values and Analysing Distribution of Data

Note In question no.3 of week 1, i have to plot frequency of given variable that is same i mean to say that is histogram only.

In [13]:

```
plt.figure(figsize=(6,4),dpi=100)
plt.xlabel('Glucose Class')
df['Glucose'].plot.hist()
sns.set_style(style='darkgrid')
print("Mean of Glucose level is :-", df['Glucose'].mean())
print("Datatype of Glucose Variable is:",df['Glucose'].dtypes)
```

```
Mean of Glucose level is :- 120.89453125
Datatype of Glucose Variable is: int64
```



I am treating missing values which is basically 0 by mean of Glucose level. This is because we can see from histogram most of observation have Glucose level between 100 and 120.

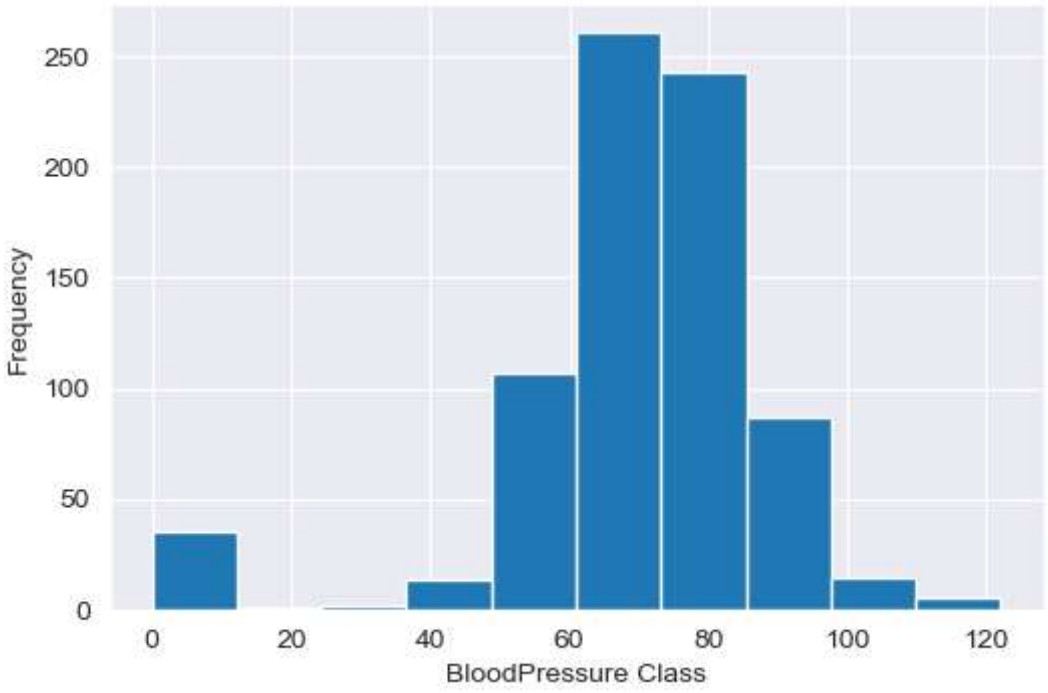
In [14]:

```
df['Glucose']=df['Glucose'].replace(0,df['Glucose'].mean())
```

In [15]:

```
plt.figure(figsize=(6,4),dpi=100)
plt.xlabel('BloodPressure Class')
df['BloodPressure'].plot.hist()
sns.set_style(style='darkgrid')
print("Mean of BloodPressure level is :-", df['BloodPressure'].mean())
print("Datatype of BloodPressure Variable is:",df['BloodPressure'].dtypes)
```

```
Mean of BloodPressure level is :- 69.10546875
Datatype of BloodPressure Variable is: int64
```



I am treating missing values which is basically 0 by mean of BloodPressure level. This is because we can see from histogram most of observation have BP level between 70 and 80.

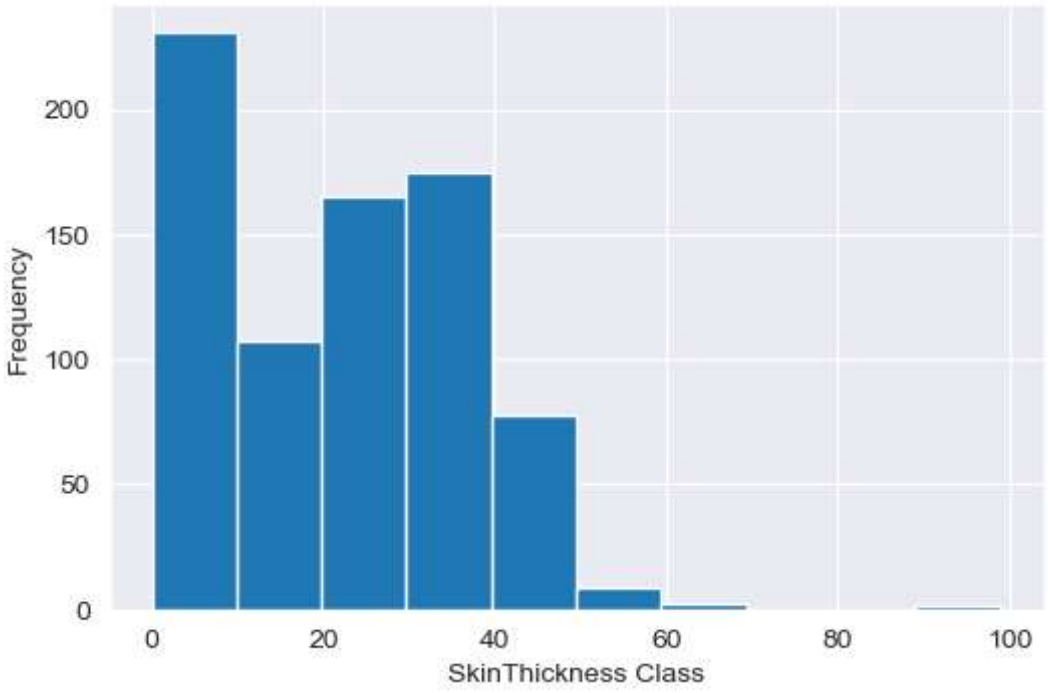
In [16]:

```
df['BloodPressure']=df['BloodPressure'].replace(0,df['BloodPressure'].mean())
```

In [17]:

```
plt.figure(figsize=(6,4),dpi=100)
plt.xlabel('SkinThickness Class')
df['SkinThickness'].plot.hist()
sns.set_style(style='darkgrid')
print("Mean of SkinThickness is :-", df['SkinThickness'].mean())
print("Datatype of SkinThickness Variable is:",df['SkinThickness'].dtypes)
```

```
Mean of SkinThickness is :- 20.536458333333332
Datatype of SkinThickness Variable is: int64
```



I am treating missing values which is basically 0 by mean of SkinThickness. This is because we can see from histogram most of observation have SkinThickness between 20 and 30.

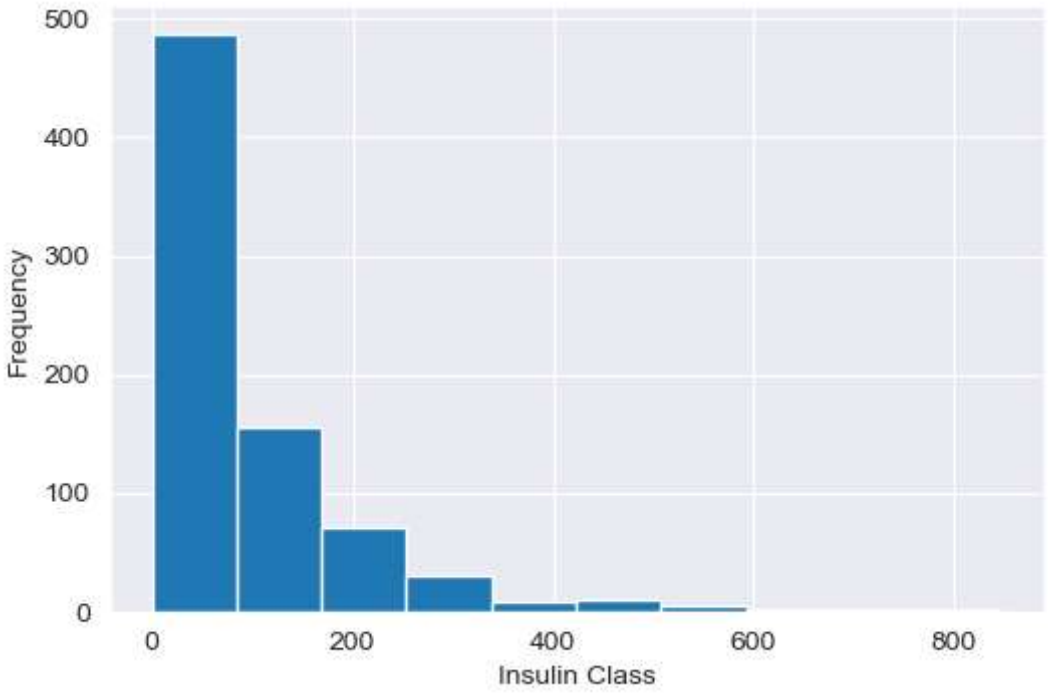
In [18]:

```
df['SkinThickness']=df['SkinThickness'].replace(0,df['SkinThickness'].mean())
```

In [19]:

```
plt.figure(figsize=(6,4),dpi=100)
plt.xlabel('Insulin Class')
df['Insulin'].plot.hist()
sns.set_style(style='darkgrid')
print("Mean of Insulin is :-", df['Insulin'].mean())
print("Datatype of Insulin Variable is:",df['Insulin'].dtypes)
```

```
Mean of Insulin is :- 79.79947916666667
Datatype of Insulin Variable is: int64
```



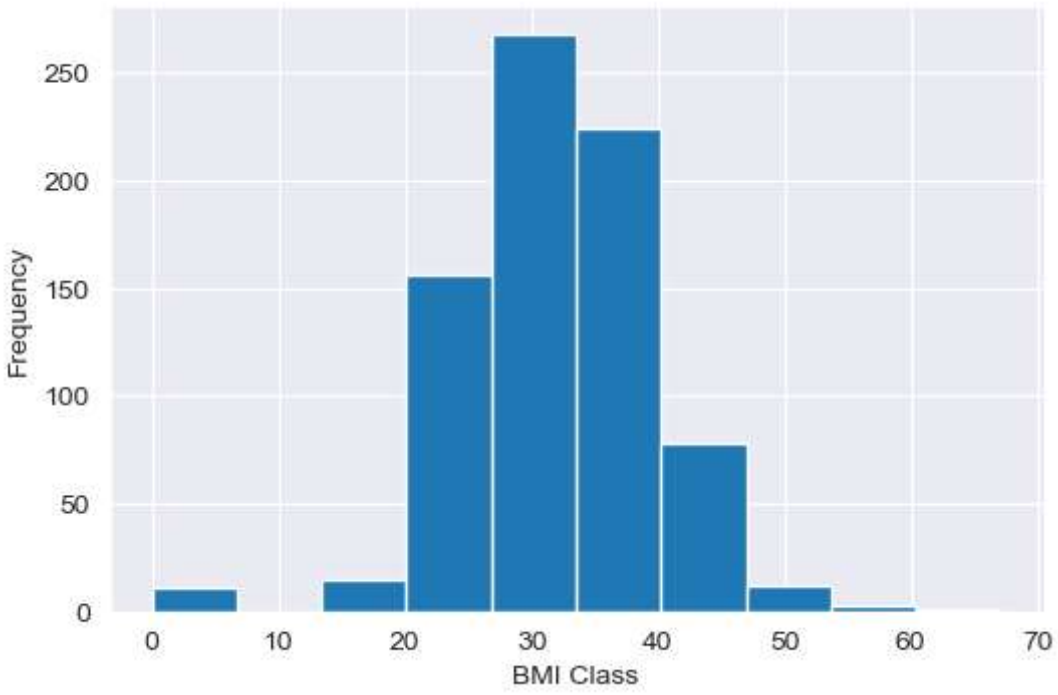
In [20]:

```
df['Insulin']=df['Insulin'].replace(0,df['Insulin'].mean())
```

In [21]:

```
plt.figure(figsize=(6,4),dpi=100)
plt.xlabel('BMI Class')
df['BMI'].plot.hist()
sns.set_style(style='darkgrid')
print("Mean of BMI is :-", df['BMI'].mean())
print("Datatype of BMI Variable is:",df['BMI'].dtypes)
```

```
Mean of BMI is :- 31.992578124999977
Datatype of BMI Variable is: float64
```



In [22]:

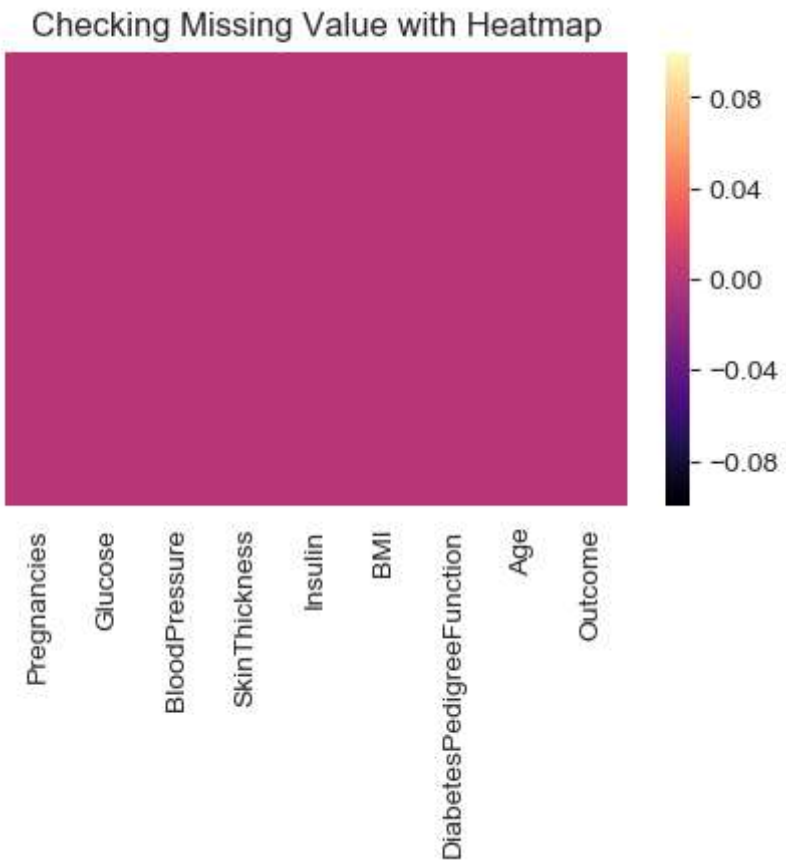
```
df['BMI']=df['BMI'].replace(0,df['BMI'].mean())
```

In [23]:

```
plt.figure(figsize=(5,3),dpi=100)
plt.title('Checking Missing Value with Heatmap')
sns.heatmap(df.isnull(),cmap='magma',yticklabels=False)
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26406fefe10>
```



In [24]:

```
df.head()
```

Out[24]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeF
0	6	148.0	72.0	35.000000	79.799479	33.6	
1	1	85.0	66.0	29.000000	79.799479	26.6	
2	8	183.0	64.0	20.536458	79.799479	23.3	
3	1	89.0	66.0	23.000000	94.000000	28.1	
4	0	137.0	40.0	35.000000	168.000000	43.1	

In [25]:

```
df.tail()
```

Out[25]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigre
763	10	101.0	76.0	48.000000	180.000000	32.9	
764	2	122.0	70.0	27.000000	79.799479	36.8	
765	5	121.0	72.0	23.000000	112.000000	26.2	
766	1	126.0	60.0	20.536458	79.799479	30.1	
767	1	93.0	70.0	31.000000	79.799479	30.4	

In [50]:

```
df.to_csv('after_week1.csv',index=False)
```