NAME:HARSH ARORA
ROLL NO:AE-1218
COURSE:BSC(HONS.) COMPUTER SCIENCE

In [3]:
```
import pandas as pd
EmployeeData = {'EMP ID':[1,2,3,4,5,6,7,8],
                'EMP NAME':['Satish','Reeya','Jay','Rahul','Roy','Jay','Vishal','Serah'],
                'SALARY': [50000,75000,100000,None,45000,100000,None,55000],
                'START DATE':['1-11-2017','12-5-2016','22-9-2015','11-10-2016','8-1-2017','22-9-2015','5-1-2016','6-2-2018']}
EMPLOYEE_DATA = pd.DataFrame(EmployeeData)
#changing the start date data from string to yyyy/mm/dd type date

EMPLOYEE_DATA['START DATE']=pd.to_datetime(EMPLOYEE_DATA['START DATE'])
print(EMPLOYEE_DATA)
```

```
   EMP ID EMP NAME    SALARY START DATE
0       1   Satish   50000.0 2017-01-11
1       2    Reeya   75000.0 2016-12-05
2       3      Jay  100000.0 2015-09-22
3       4    Rahul       NaN 2016-11-10
4       5      Roy   45000.0 2017-08-01
5       6      Jay  100000.0 2015-09-22
6       7   Vishal       NaN 2016-05-01
7       8    Serah   55000.0 2018-06-02
```

In [43]:
```
#Display the column names and the number of records.
COLUMNcount=EMPLOYEE_DATA.count()#gives no. of non-empty values in each column
print(COLUMNcount)
```

```
EMP ID        8
EMP NAME      8
SALARY        6
START DATE    8
dtype: int64
```

In [36]:
```
#Display the first 4 records of the dataset.
print(EMPLOYEE_DATA.head(4))
```

```
   EMP ID EMP NAME    SALARY START DATE
0       1   Satish   50000.0 2017-01-11
1       2    Reeya   75000.0 2016-12-05
2       3      Jay  100000.0 2015-09-22
3       4    Rahul       NaN 2016-11-10
```

In [16]:
```
#For each numeric attribute, evaluate various statistical parameters using describe() function
print(EMPLOYEE_DATA.describe())
#CAN ALSO USE print(EMPLOYEE_DATA['SALARY'].describe()) to get result only for salary
```

```
        EMP ID         SALARY
count  8.00000       6.000000
mean   4.50000   70833.333333
std    2.44949   24782.386218
min    1.00000   45000.000000
25%    2.75000   51250.000000
50%    4.50000   65000.000000
75%    6.25000   93750.000000
max    8.00000  100000.000000
```

In [10]:
```
#Check for the presence of missing values in the dataset and replace them with some valid numeric value
print(EMPLOYEE_DATA.isnull())
EMPLOYEE_DATA_filled=EMPLOYEE_DATA.fillna(50000)
print('Original data table\n',EMPLOYEE_DATA)
print()
print('filled data table\n',EMPLOYEE_DATA_filled)
```

```
   EMP ID  EMP NAME  SALARY  START DATE
0   False     False   False       False
1   False     False   False       False
2   False     False   False       False
3   False     False    True       False
4   False     False   False       False
5   False     False   False       False
6   False     False    True       False
7   False     False   False       False
Original data table
    EMP ID EMP NAME    SALARY START DATE
0       1   Satish   50000.0 2017-01-11
1       2    Reeya   75000.0 2016-12-05
2       3      Jay  100000.0 2015-09-22
3       4    Rahul       NaN 2016-11-10
4       5      Roy   45000.0 2017-08-01
5       6      Jay  100000.0 2015-09-22
6       7   Vishal       NaN 2016-05-01
7       8    Serah   55000.0 2018-06-02

filled data table
    EMP ID EMP NAME    SALARY START DATE
0       1   Satish   50000.0 2017-01-11
1       2    Reeya   75000.0 2016-12-05
2       3      Jay  100000.0 2015-09-22
3       4    Rahul   50000.0 2016-11-10
4       5      Roy   45000.0 2017-08-01
5       6      Jay  100000.0 2015-09-22
6       7   Vishal   50000.0 2016-05-01
7       8    Serah   55000.0 2018-06-02
```

In [8]:
```
#Find and remove duplicate records (if any) in the dataset.
print(EMPLOYEE_DATA.duplicated())
EMPLOYEE_DATA_clean=EMPLOYEE_DATA.dropna()
print('Original data table\n',EMPLOYEE_DATA)
print()
print('filled data table\n',EMPLOYEE_DATA_clean)
```

```
0    False
1    False
2    False
3    False
4    False
5    False
6    False
7    False
dtype: bool
Original data table
    EMP ID EMP NAME    SALARY START DATE
0       1   Satish   50000.0 2017-01-11
1       2    Reeya   75000.0 2016-12-05
2       3      Jay  100000.0 2015-09-22
3       4    Rahul       NaN 2016-11-10
4       5      Roy   45000.0 2017-08-01
5       6      Jay  100000.0 2015-09-22
6       7   Vishal       NaN 2016-05-01
7       8    Serah   55000.0 2018-06-02

filled data table
    EMP ID EMP NAME    SALARY START DATE
0       1   Satish   50000.0 2017-01-11
1       2    Reeya   75000.0 2016-12-05
2       3      Jay  100000.0 2015-09-22
4       5      Roy   45000.0 2017-08-01
5       6      Jay  100000.0 2015-09-22
7       8    Serah   55000.0 2018-06-02
```

In [4]:
```
#Pima Indians Diabetes Dataset
DIABETES_DATA = pd.read_csv(r"C:\Users\HP\Downloads\diabetes.csv")
```

In [47]:
```
#Display the column names and the number of records.
print(DIABETES_DATA.count())
```

```
Pregnancies                 768
Glucose                     768
BloodPressure               768
SkinThickness               768
Insulin                     768
BMI                         768
DiabetesPedigreeFunction    768
Age                         768
Outcome                     768
dtype: int64
```

In [7]:
```
#Display the first 10 records of the dataset.
print(DIABETES_DATA.head(10))
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1
5            5      116             74              0        0  25.6
6            3       78             50             32       88  31.0
7           10      115              0              0        0  35.3
8            2      197             70             45      543  30.5
9            8      125             96              0        0   0.0

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
5                     0.201   30        0
6                     0.248   26        1
7                     0.134   29        0
8                     0.158   53        1
9                     0.232   54        1
```

In [49]:
```
#For each numeric attribute, evaluate various statistical parameters using describe() function
print(DIABETES_DATA.describe())
```

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

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    31.992578                  0.471876   33.240885    0.348958
std      7.884160                  0.331329   11.760232    0.476951
min      0.000000                  0.078000   21.000000    0.000000
25%     27.300000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000
```

In [5]:
```
#Check for the presence of missing values in the dataset and replace them with some valid numeric value
print(DIABETES_DATA.isnull())
DIABETES_DATA_filled=DIABETES_DATA.fillna(100)
print('Original data table\n',DIABETES_DATA)
print()
print('filled data table\n',DIABETES_DATA_filled)
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin    BMI  \
0          False    False          False          False    False  False
1          False    False          False          False    False  False
2          False    False          False          False    False  False
3          False    False          False          False    False  False
4          False    False          False          False    False  False
..           ...      ...            ...            ...      ...    ...
763        False    False          False          False    False  False
764        False    False          False          False    False  False
765        False    False          False          False    False  False
766        False    False          False          False    False  False
767        False    False          False          False    False  False

     DiabetesPedigreeFunction    Age  Outcome
0                       False  False    False
1                       False  False    False
2                       False  False    False
3                       False  False    False
4                       False  False    False
..                        ...    ...      ...
763                     False  False    False
764                     False  False    False
765                     False  False    False
766                     False  False    False
767                     False  False    False

[768 rows x 9 columns]
Original data table
      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0               6      148             72             35        0  33.6
1               1       85             66             29        0  26.6
2               8      183             64              0        0  23.3
3               1       89             66             23       94  28.1
4               0      137             40             35      168  43.1
..            ...      ...            ...            ...      ...   ...
763            10      101             76             48      180  32.9
764             2      122             70             27        0  36.8
765             5      121             72             23      112  26.2
766             1      126             60              0        0  30.1
767             1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]

filled data table
      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0               6      148             72             35        0  33.6
1               1       85             66             29        0  26.6
2               8      183             64              0        0  23.3
3               1       89             66             23       94  28.1
4               0      137             40             35      168  43.1
..            ...      ...            ...            ...      ...   ...
763            10      101             76             48      180  32.9
764             2      122             70             27        0  36.8
765             5      121             72             23      112  26.2
766             1      126             60              0        0  30.1
767             1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]
```

In [6]:
```
#Find and remove duplicate records (if any) in the dataset.
print(DIABETES_DATA.duplicated())
DIABETES_DATA_clean=DIABETES_DATA.dropna()
print('Original data table\n',DIABETES_DATA)
print()
print('filled data table\n',DIABETES_DATA_clean)
```

```
0      False
1      False
2      False
3      False
4      False
       ...
763    False
764    False
765    False
766    False
767    False
Length: 768, dtype: bool
Original data table
      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0               6      148             72             35        0  33.6
1               1       85             66             29        0  26.6
2               8      183             64              0        0  23.3
3               1       89             66             23       94  28.1
4               0      137             40             35      168  43.1
..            ...      ...            ...            ...      ...   ...
763            10      101             76             48      180  32.9
764             2      122             70             27        0  36.8
765             5      121             72             23      112  26.2
766             1      126             60              0        0  30.1
767             1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]

filled data table
      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0               6      148             72             35        0  33.6
1               1       85             66             29        0  26.6
2               8      183             64              0        0  23.3
3               1       89             66             23       94  28.1
4               0      137             40             35      168  43.1
..            ...      ...            ...            ...      ...   ...
763            10      101             76             48      180  32.9
764             2      122             70             27        0  36.8
765             5      121             72             23      112  26.2
766             1      126             60              0        0  30.1
767             1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]
```

In [57]:
```
#Show scatter plot depicting relationship between two numeric columns of your choice.
# Scatter Plot between Current Vs Voltage from the given dataset

import pandas as pd
import matplotlib.pyplot as plt

DATA = pd.read_csv((r"C:\Users\HP\Downloads\diabetes.csv"))
DATA.plot(kind='scatter', x='Age', y='BloodPressure',marker='X')
plt.show()
```