Importing Required Libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style("whitegrid")
```

Reading the CSV file from Local Disk

In [2]:

```
df = pd.read_csv("Student dataset/student_performance.csv")
```

Checking number of Colums and Rows in the Dataset

In [3]:

```
print(df.head(10))
```

```
    RegID          School  Gender  Age Address Family Size Pstatus  \
0  110091  Gabriel Pereira  Female   18   Urban         > 3       A
1  110092  Gabriel Pereira  Female   17   Urban         > 3       T
2  110093  Gabriel Pereira  Female   15   Urban         < 3       T
3  110094  Gabriel Pereira  Female   15   Urban         > 3       T
4  110095  Gabriel Pereira  Female   16   Urban         > 3       T
5  110096  Gabriel Pereira    Male   16   Urban         < 3       T
6  110097  Gabriel Pereira    Male   16   Urban         < 3       T
7  110098  Gabriel Pereira  Female   17   Urban         > 3       A
8  110099  Gabriel Pereira    Male   15   Urban         < 3       A
9  110100  Gabriel Pereira    Male   15   Urban         > 3       T

    Mother Education  Fathers Education Mother's Job  ... Romantic FamRel  \
0  Bachelor's Degree  Bachelor's Degree      at_home  ...       no    4.0
1              Other              Other      at_home  ...       no    5.0
2              Other              Other      at_home  ...       no    4.0
3  Bachelor's Degree            10/10+2       health  ...      yes    3.0
4            10/10+2            10/10+2        other  ...       no    4.0
5  Bachelor's Degree            10/10+2     services  ...       no    5.0
6            10/10+2            10/10+2        other  ...       no    4.0
7  Bachelor's Degree  Bachelor's Degree        other  ...       no    NaN
8            10/10+2            10/10+2     services  ...       no    4.0
9            10/10+2  Bachelor's Degree        other  ...       no    5.0

  FreeTime  GoOut Health  Absences Language  Science Maths  Percentage
0      3.0    4.0    3.0       6.0      25.0       30  19.2   24.733333
1      3.0    3.0    3.0       4.0      25.0       25  19.2   23.066667
2      3.0    2.0    3.0      10.0      35.0       40  32.0   35.666667
3      2.0    2.0    5.0       2.0      75.0       70  48.0   64.333333
4      3.0    2.0    5.0       4.0      30.0       50  32.0   37.333333
5      4.0    2.0    5.0      10.0      75.0       75  48.0   66.000000
6      4.0    4.0    3.0       0.0      60.0       60  35.2   51.733333
7      NaN    4.0    1.0       6.0      30.0       25  19.2   24.733333
8      2.0    2.0    1.0       0.0      80.0       90  60.8   76.933333
9      5.0    1.0    5.0       0.0      70.0       75  48.0   64.333333

[10 rows x 33 columns]
```

In [4]:

```
print(df.tail(10))
```

```
      RegID                School  Gender  Age Address Family Size Pstatus  \
385  110476  Mousinho da Silveira  Female   18   Rural         > 3       T   
386  110477  Mousinho da Silveira  Female   18   Rural         > 3       T   
387  110478  Mousinho da Silveira  Female   19   Rural         > 3       T   
388  110479  Mousinho da Silveira  Female   18   Urban         < 3       T   
389  110480  Mousinho da Silveira  Female   18   Urban         > 3       T   
390  110481  Mousinho da Silveira    Male   20   Urban         < 3       A   
391  110482  Mousinho da Silveira    Male   17   Urban         < 3       T   
392  110483  Mousinho da Silveira    Male   21   Rural         > 3       T   
393  110484  Mousinho da Silveira    Male   18   Rural         < 3       T   
394  110485  Mousinho da Silveira    Male   19   Urban         < 3       T   

      Mother Education  Fathers Education Mother's Job  ... Romantic FamRel  \
385            10/10+2            10/10+2      at_home  ...       no    5.0   
386  Bachelor's Degree  Bachelor's Degree      teacher  ...      yes    4.0   
387            10/10+2            10/10+2     services  ...       no    5.0   
388            10/10+2              Other      teacher  ...       no    4.0   
389              Other              Other        other  ...       no    1.0   
390            10/10+2            10/10+2     services  ...       no    5.0   
391            10/10+2              Other     services  ...       no    2.0   
392              Other              Other        other  ...       no    5.0   
393            10/10+2            10/10+2     services  ...       no    4.0   
394              Other              Other        other  ...       no    3.0   

    FreeTime  GoOut Health  Absences Language Science Maths Percentage  
385      3.0    3.0    4.0       2.0     50.0      45  32.0  42.333333  
386      4.0    3.0    5.0       7.0     30.0      25  19.2  24.733333  
387      4.0    2.0    5.0       0.0     35.0      25   0.0  20.000000  
388      3.0    4.0    1.0       0.0     35.0      45  25.6  35.200000  
389      1.0    1.0    5.0       0.0     30.0      25   0.0  18.333333  
390      5.0    4.0    4.0      11.0     45.0      45  28.8  39.600000  
391      4.0    5.0    2.0       3.0     70.0      80  51.2  67.066667  
392      5.0    3.0    3.0       3.0     50.0      40  22.4  37.466667  
393      4.0    1.0    5.0       0.0     55.0      60  32.0  49.000000  
394      2.0    3.0    5.0       5.0     40.0      45  28.8  37.933333  

[10 rows x 33 columns]
```

As we can see here in our dataset there are total 33 Columns and 394 rows

---

Checking the information of Each Columns

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 33 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   RegID              395 non-null    int64  
 1   School             395 non-null    object 
 2   Gender             395 non-null    object 
 3   Age                395 non-null    int64  
 4   Address            395 non-null    object 
 5   Family Size        395 non-null    object 
 6   Pstatus            395 non-null    object 
 7   Mother Education   395 non-null    object 
 8   Fathers Education  395 non-null    object 
 9   Mother's Job       395 non-null    object 
 10  Father's Job       395 non-null    object 
 11  Reason             395 non-null    object 
 12  Guardian           395 non-null    object 
 13  Travel time        395 non-null    int64  
 14  Study Time         395 non-null    object 
 15  Failures           385 non-null    float64
 16  School Support     395 non-null    object 
 17  Family Support     395 non-null    object 
 18  Paid               395 non-null    object 
 19  Activities         395 non-null    object 
 20  Nursery            395 non-null    object 
 21  Higher             395 non-null    object 
 22  Internet           395 non-null    object 
 23  Romantic           395 non-null    object 
 24  FamRel             379 non-null    float64
 25  FreeTime           385 non-null    float64
 26  GoOut              385 non-null    float64
 27  Health             388 non-null    float64
 28  Absences           387 non-null    float64
 29  Language           394 non-null    float64
 30  Science            395 non-null    int64  
 31  Maths              395 non-null    float64
 32  Percentage         395 non-null    float64
dtypes: float64(9), int64(4), object(20)
memory usage: 102.0+ KB
```

Function is used to generate descriptive statistics like mean, median, mode, standard deviation

In [6]:
```
df.describe()
```

Out[6]:

	RegID	Age	Travel time	Failures	FamRel	FreeTime	GoOut	Health	Absences	Language	Science	Ma
count	395.000000	395.000000	395.000000	385.000000	379.000000	385.000000	385.000000	388.000000	387.000000	394.000000	395.000000	395.000
mean	110288.000000	16.696203	1.448101	0.342857	3.944591	3.238961	3.109091	3.543814	5.764858	54.593909	53.569620	33.328
std	114.170924	1.276043	0.697505	0.751289	0.896549	0.994798	1.110342	1.392352	8.067012	16.587728	18.807523	14.660
min	110091.000000	15.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	0.000000	15.000000	0.000000	0.000
25%	110189.500000	16.000000	1.000000	0.000000	4.000000	3.000000	2.000000	3.000000	0.000000	40.000000	45.000000	25.600
50%	110288.000000	17.000000	1.000000	0.000000	4.000000	3.000000	3.000000	4.000000	4.000000	55.000000	55.000000	35.200
75%	110386.500000	18.000000	2.000000	0.000000	5.000000	4.000000	4.000000	5.000000	8.000000	65.000000	65.000000	44.800
max	110485.000000	22.000000	4.000000	3.000000	5.000000	5.000000	5.000000	5.000000	75.000000	95.000000	95.000000	64.000

Checking the datatypes of Each Columns

In [7]:
```
df.dtypes
```

Out[7]:
```
RegID                  int64
School                object
Gender                object
Age                    int64
Address               object
Family Size           object
Pstatus               object
Mother Education      object
Fathers Education     object
Mother's Job          object
Father's Job          object
Reason                object
Guardian              object
Travel time            int64
Study Time            object
Failures             float64
School Support        object
Family Support        object
Paid                  object
Activities            object
Nursery               object
Higher                object
Internet              object
Romantic              object
FamRel               float64
FreeTime             float64
GoOut                float64
Health               float64
Absences             float64
Language             float64
Science                int64
Maths                float64
Percentage           float64
dtype: object
```

Checking the size of Dataset

In [8]:
```
size = df.size
print("Size = {}".format(size))
```

```
Size = 13035
```

Checking shape of dataset

In [9]:
```
shape = df.shape
print("Shape = {}".format(shape))
```

```
Shape = (395, 33)
```

Checking the name of all coloumns in dataset

In [10]:

```
columns = df.columns
print("columns = {}".format(columns))
```

```
columns = Index(['RegID', 'School', 'Gender', 'Age', 'Address', 'Family Size', 'Pstatus',
       'Mother Education', 'Fathers Education', 'Mother's Job', 'Father's Job',
       'Reason', 'Guardian', 'Travel time', 'Study Time', 'Failures',
       'School Support', 'Family Support', 'Paid', 'Activities', 'Nursery',
       'Higher', 'Internet', 'Romantic', 'FamRel', 'FreeTime', 'GoOut',
       'Health', 'Absences', 'Language', 'Science', 'Maths', 'Percentage'],
      dtype='object')
```

Checking Unique values of different Columns

In [11]:

```
df['Gender'].unique()
```

Out[11]:

```
array(['Female', 'Male'], dtype=object)
```

In [12]:

```
df['School'].unique()
```

Out[12]:

```
array(['Gabriel Pereira', 'Mousinho da Silveira'], dtype=object)
```

In [13]:

```
df['Address'].unique()
```

Out[13]:

```
array(['Urban', 'Rural'], dtype=object)
```

In [14]:

```
df['Family Size'].unique()
```

Out[14]:

```
array(['> 3', '< 3'], dtype=object)
```

In [15]:

```
df['Mother Education'].unique()
```

Out[15]:

```
array(["Bachelor's Degree", 'Other', '10/10+2', "Master's Degree"],
      dtype=object)
```

In [16]:

```
df['Fathers Education'].unique()
```

Out[16]:

```
array(["Bachelor's Degree", 'Other', '10/10+2', "Master's Degree"],
      dtype=object)
```

In [17]:

```
df["Mother's Job"].unique()
```

Out[17]:

```
array(['at_home', 'health', 'other', 'services', 'teacher'], dtype=object)
```

In [18]:

```
df["Father's Job"].unique()
```

Out[18]:

```
array(['teacher', 'other', 'services', 'health', 'at_home'], dtype=object)
```

In [19]:

```
df["Reason"].unique()
```

Out[19]:

```
array(['course', 'other', 'home', 'reputation'], dtype=object)
```

In [20]:

```
df["Family Size"].unique()
```

Out[20]:

```
array(['> 3', '< 3'], dtype=object)
```

Checking Null Values

In [21]:

```
df.isnull().sum()
```

Out[21]:

```
RegID                 0
School                0
Gender                0
Age                   0
Address               0
Family Size           0
Pstatus               0
Mother Education      0
Fathers Education     0
Mother's Job          0
Father's Job          0
Reason                0
Guardian              0
Travel time           0
Study Time            0
Failures             10
School Support        0
Family Support        0
Paid                  0
Activities            0
Nursery               0
Higher                0
Internet              0
Romantic              0
FamRel               16
FreeTime             10
GoOut                10
Health                7
Absences              8
Language              1
Science               0
Maths                 0
Percentage            0
dtype: int64
```

There are 10 null values in Failures column, 16 null values in FamRel Column, 10 null values in FreeTime Column 7 null values in Health Column, 8 null values in Absenses column and 1 null value in Language column.

Data Cleaning

Replacing all null values with 0 in Failure column

In [22]:

```
df["Failures"].isnull().sum()
```

Out[22]:

```
10
```

In [23]:

```
df['Failures'] = df['Failures'].fillna(0)
```

In [24]:

```
df["Failures"].isnull().sum()
```

Out[24]:

```
0
```

Replacing all null values in Family Relation colum with mean values and applied floor of that column to remove the decimal values

In [25]:

```
df["FamRel"].isnull().sum()
```

Out[25]:

```
16
```

In [26]:

```
averageFamilyRelation = df['FamRel'].mean()
df['FamRel'] = df['FamRel'].fillna(averageFamilyRelation)
df['FamRel'] = df['FamRel'].apply(np.floor)
```

In [27]:

```
df["FamRel"].isnull().sum()
```

Out[27]:

```
0
```

Replacing all null values in FreeTime colum with mean values and applied floor of that column to remove the decimal values

In [28]:

```
df["FreeTime"].isnull().sum()
```

Out[28]:

```
10
```

In [29]:

```
averageFreeTime = df['FreeTime'].mean()
df['FreeTime'] = df['FreeTime'].fillna(averageFreeTime)
df['FreeTime'] = df['FreeTime'].apply(np.floor)
```

In [30]:

```
df["FreeTime"].isnull().sum()
```

Out[30]:

```
0
```

Replacing all null values in GoOut colum with mean values and applied floor of that column to remove the decimal values

In [31]:

```
df["GoOut"].isnull().sum()
```

Out[31]:

```
10
```

In [32]:

```
averageGoOut = df['GoOut'].mean()
df['GoOut'] = df['GoOut'].fillna(averageGoOut)
df['GoOut'] = df['GoOut'].apply(np.floor)
```

In [33]:

```
df["GoOut"].isnull().sum()
```

Out[33]:

```
0
```

Replacing all null values in Health colum with median value of that column

In [34]:

```
df["Health"].isnull().sum()
```

Out[34]:

```
7
```

In [35]:

```
medianOfHealth = df['Health'].median()
df['Health'] = df['Health'].fillna(medianOfHealth)
```

In [36]:

```
df["Health"].isnull().sum()
```

Out[36]:

```
0
```

Replacing all null values in Absences colum with median value of that column

In [37]:

```
df["Absences"].isnull().sum()
```

Out[37]:

```
8
```

In [38]:

```
medianOfAbsenses = df['Absences'].median()
df['Absences'] = df['Absences'].fillna(medianOfAbsenses)
```

In [39]:

```
df["Absences"].isnull().sum()
```

Out[39]:

```
0
```

Replacing all null values in Language colum with mean value of that column

In [40]:

```
df["Language"].isnull().sum()
```

Out[40]:

```
1
```

In [41]:

```
averageLanguage = df['Language'].mean()
df['Language'] = df['Language'].fillna(averageLanguage)
```

In [42]:

```
df["Language"].isnull().sum()
```

Out[42]:

```
0
```

Checking Null Values Again

In [43]:

```
df.isnull().sum()
```

Out[43]:

```
RegID                0
School               0
Gender               0
Age                  0
Address              0
Family Size          0
Pstatus              0
Mother Education     0
Fathers Education    0
Mother's Job         0
Father's Job         0
Reason               0
Guardian             0
Travel time          0
Study Time           0
Failures             0
School Support       0
Family Support       0
Paid                 0
Activities           0
Nursery              0
Higher               0
Internet             0
Romantic             0
FamRel               0
FreeTime             0
GoOut                0
Health               0
Absences             0
Language             0
Science              0
Maths                0
Percentage           0
dtype: int64
```

Our dataset is now clean and ready for analysis

Data Wrangling

Applying Filter

In [44]:

```
ndFrame=df[
(df["FamRel"]>3) &
(df["FreeTime"]>4) &
(df["GoOut"]>3)
]
print(df)
```

```
       RegID                School  Gender  Age Address Family Size Pstatus  \
0     110091       Gabriel Pereira  Female   18   Urban         > 3       A
1     110092       Gabriel Pereira  Female   17   Urban         > 3       T
2     110093       Gabriel Pereira  Female   15   Urban         < 3       T
3     110094       Gabriel Pereira  Female   15   Urban         > 3       T
4     110095       Gabriel Pereira  Female   16   Urban         > 3       T
..       ...                   ...     ...  ...     ...         ...     ...
390   110481  Mousinho da Silveira    Male   20   Urban         < 3       A
391   110482  Mousinho da Silveira    Male   17   Urban         < 3       T
392   110483  Mousinho da Silveira    Male   21   Rural         > 3       T
393   110484  Mousinho da Silveira    Male   18   Rural         < 3       T
394   110485  Mousinho da Silveira    Male   19   Urban         < 3       T

       Mother Education  Fathers Education Mother's Job  ... Romantic FamRel  \
0     Bachelor's Degree  Bachelor's Degree      at_home  ...       no    4.0
1                 Other              Other      at_home  ...       no    5.0
2                 Other              Other      at_home  ...       no    4.0
3     Bachelor's Degree            10/10+2       health  ...      yes    3.0
4               10/10+2            10/10+2        other  ...       no    4.0
..                  ...                ...          ...  ...      ...    ...
390             10/10+2            10/10+2     services  ...       no    5.0
391             10/10+2              Other     services  ...       no    2.0
392               Other              Other        other  ...       no    5.0
393             10/10+2            10/10+2     services  ...       no    4.0
394               Other              Other        other  ...       no    3.0

     FreeTime  GoOut  Health  Absences  Language  Science  Maths  Percentage
0         3.0    4.0     3.0       6.0      25.0       30   19.2   24.733333
1         3.0    3.0     3.0       4.0      25.0       25   19.2   23.066667
2         3.0    2.0     3.0      10.0      35.0       40   32.0   35.666667
3         2.0    2.0     5.0       2.0      75.0       70   48.0   64.333333
4         3.0    2.0     5.0       4.0      30.0       50   32.0   37.333333
..        ...    ...     ...       ...       ...      ...    ...         ...
390       5.0    4.0     4.0      11.0      45.0       45   28.8   39.600000
391       4.0    5.0     2.0       3.0      70.0       80   51.2   67.066667
392       5.0    3.0     3.0       3.0      50.0       40   22.4   37.466667
393       4.0    1.0     5.0       0.0      55.0       60   32.0   49.000000
394       2.0    3.0     5.0       5.0      40.0       45   28.8   37.933333

[395 rows x 33 columns]
```

Through this example we are filtering Students having Family Relation greater than 3 and students who got free times more then 4 hours and those students who go out of house for more then 3 hours

---

Finding the students who scored highest marks in Science

In [45]:

```
highestScience = df["Science"].max()
hdf=df[df["Science"]==highestScience]
hdf
```

Out[45]:

	RegID	School	Gender	Age	Address	Family Size	Pstatus	Mother Education	Fathers Education	Mother's Job	...	Romantic	FamRel	FreeTime	GoOut	Health	Abs
47	110138	Gabriel Pereira	Male	16	Urban	> 3	T	Bachelor's Degree	10/10+2	health	...	no	4.0	2.0	2.0	2.0	
110	110201	Gabriel Pereira	Male	15	Urban	< 3	A	Bachelor's Degree	Bachelor's Degree	teacher	...	no	5.0	5.0	3.0	4.0	
113	110204	Gabriel Pereira	Male	15	Urban	< 3	T	Bachelor's Degree	10/10+2	teacher	...	no	3.0	5.0	2.0	3.0	

3 rows × 33 columns

These are the students who scored highest marks in Science

Finding the students who scored highest marks in Maths

In [46]:

```
highestMaths = df["Maths"].max()
hdf=df[df["Maths"]==highestMaths]
hdf
```

Out[46]:

	RegID	School	Gender	Age	Address	Family Size	Pstatus	Mother Education	Fathers Education	Mother's Job	...	Romantic	FamRel	FreeTime	GoOut	Health	Absei
47	110138	Gabriel Pereira	Male	16	Urban	> 3	T	Bachelor's Degree	10/10+2	health	...	no	4.0	2.0	2.0	2.0	

1 rows × 33 columns

These are the students who scored highest marks in Maths

Finding the students who scored highest marks in Language

In [47]:
```python
highestLanguage = df["Language"].max()
hdf=df[df["Language"]==highestLanguage
]
hdf
```

Out[47]:

	RegID	School	Gender	Age	Address	Family Size	Pstatus	Mother Education	Fathers Education	Mother's Job	...	Romantic	FamRel	FreeTime	GoOut	Health	A
42	110133	Gabriel Pereira	Male	15	Urban	> 3	T	Bachelor's Degree	Bachelor's Degree	services	...	no	4.0	3.0	3.0	5.0	
47	110138	Gabriel Pereira	Male	16	Urban	> 3	T	Bachelor's Degree	10/10+2	health	...	no	4.0	2.0	2.0	2.0	
374	110465	Mousinho da Silveira	Female	18	Rural	< 3	T	Bachelor's Degree	Bachelor's Degree	other	...	no	5.0	4.0	4.0	1.0	

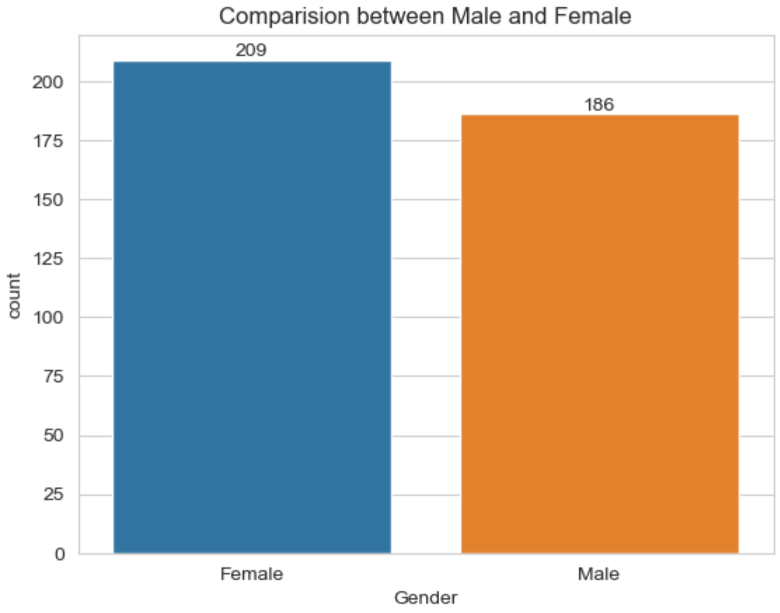
3 rows × 33 columns

These are the students who scored highest marks in Language

Data analysis and Visualization

1. Comparision between Male and Female Students

In [48]:
```python
ax = sns.countplot(data = df, x = "Gender")
ax.bar_label(ax.containers[0])
plt.title("Comparision between Male and Female")
plt.show()
```



Conclusion

From above plot we can conclude that the number of Female Students is more then the number of Male Students

---

2. School wise comparision between Male and Female Students

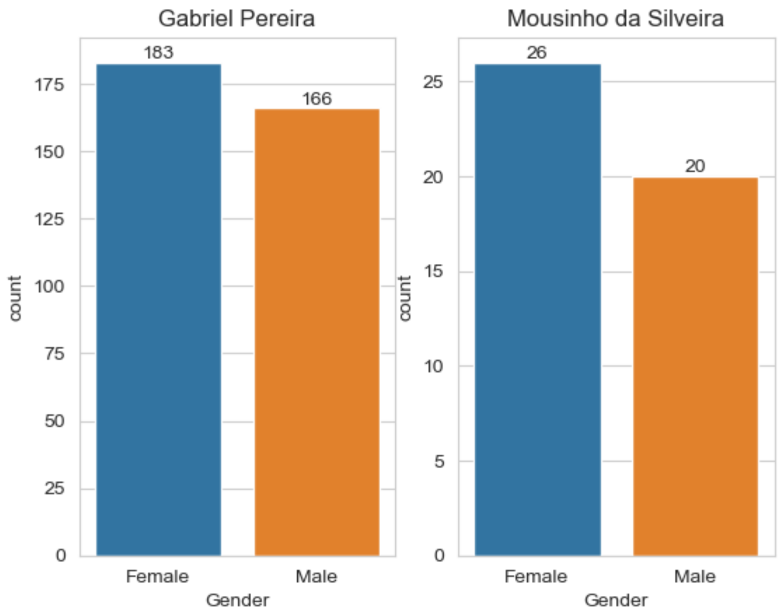
In [49]:

```
gp = df.loc[df['School'] == "Gabriel Pereira"]
ms = df.loc[df['School'] == "Mousinho da Silveira"]

gpPlot=plt.subplot(1,2,1)
gpPlot.title.set_text('Gabriel Pereira')
gpax = sns.countplot(data = gp, x = "Gender")
gpax.bar_label(gpax.containers[0])

msPlot=plt.subplot(1,2,2)
msPlot.title.set_text('Mousinho da Silveira')
msax = sns.countplot(data = ms, x = "Gender")
msax.bar_label(msax.containers[0])

plt.show()
```



Conclusion

From above analysis we can conclude that the number of Female Students is more then the number of Male Students in both the Schools

---

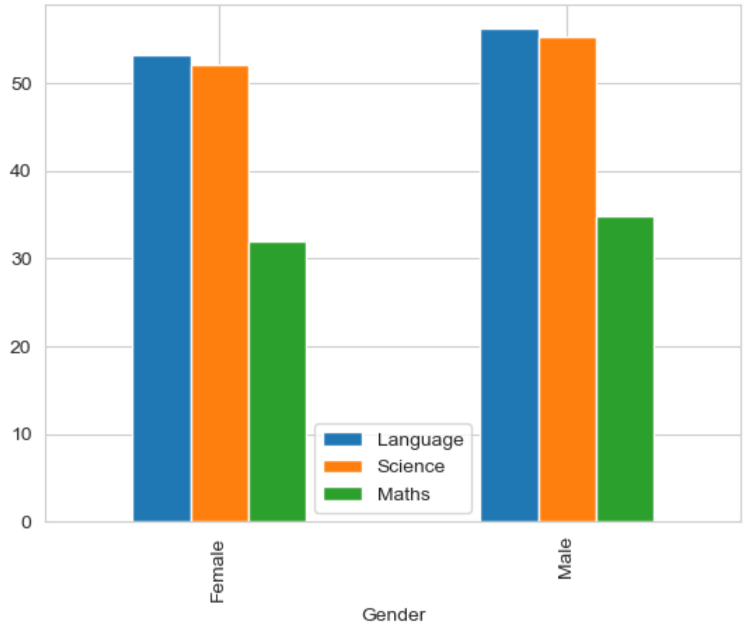
3. Comparision between Male and Female Student's Performance in each Subject

In [50]:

```
df.groupby("Gender").agg({"Language" : "mean","Science" : "mean","Maths":"mean"}).plot(kind='bar')
```

Out[50]:

```
<Axes: xlabel='Gender'>
```



Conclusion

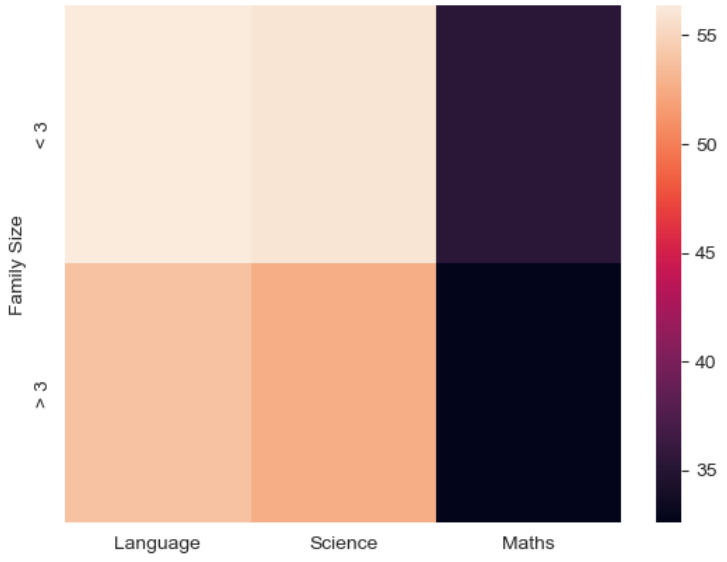
From above analysis we can conclude that male students performed little good in compare to Female Students

---

4. Effect of Family Size on Student's Academic Performance

In [51]:
```
famsize = df.groupby("Family Size").agg({"Language" : "mean","Science" : "mean","Maths":"mean"})
sns.heatmap(famsize)
plt.show
```

Out[51]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Conclusion

From the above analysis we can conclude that those students who has more family member is little bit distracted and has low performance in compare to those students who has comparatively less family size

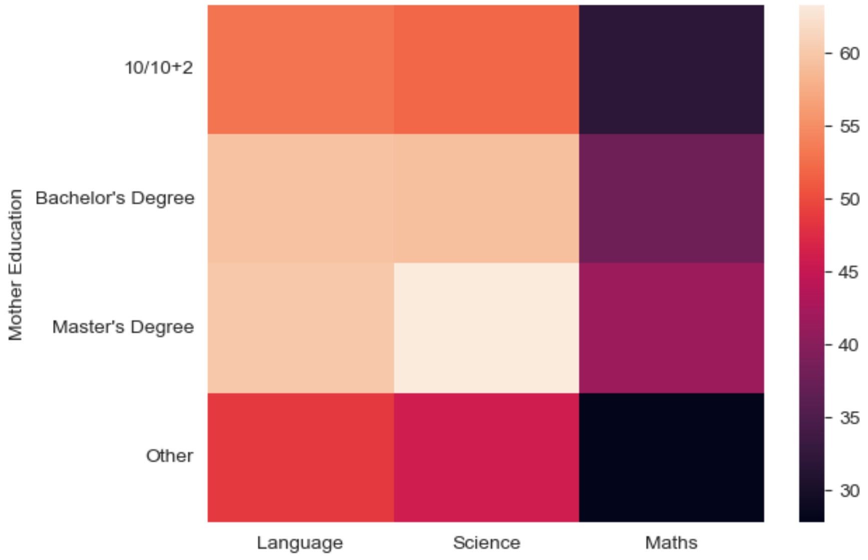
---

5. Effect of Mother's Education on Student's Academic Performance

In [52]:
```
medu = df.groupby("Mother Education").agg({"Language" : "mean","Science" : "mean","Maths":"mean"})
sns.heatmap(medu)
plt.show
```

Out[52]:

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



Conclusion

From the above analysis we can conclude that the Mother's Education has good impact on child's academic Performace

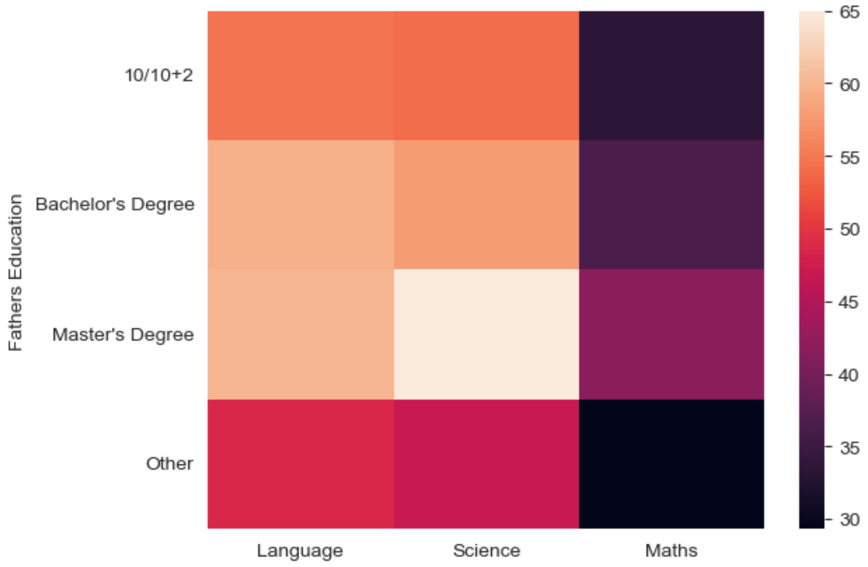
6. Effect of Father's Education on Student's Academic Performance

In [53]:

```
fedu = df.groupby("Fathers Education").agg({"Language" : "mean","Science" : "mean","Maths":"mean"})
sns.heatmap(fedu)
plt.show
```

Out[53]:

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



Conclusion

From the above analysis we can conclude that the Father's Education has good impact on child's academic Performace

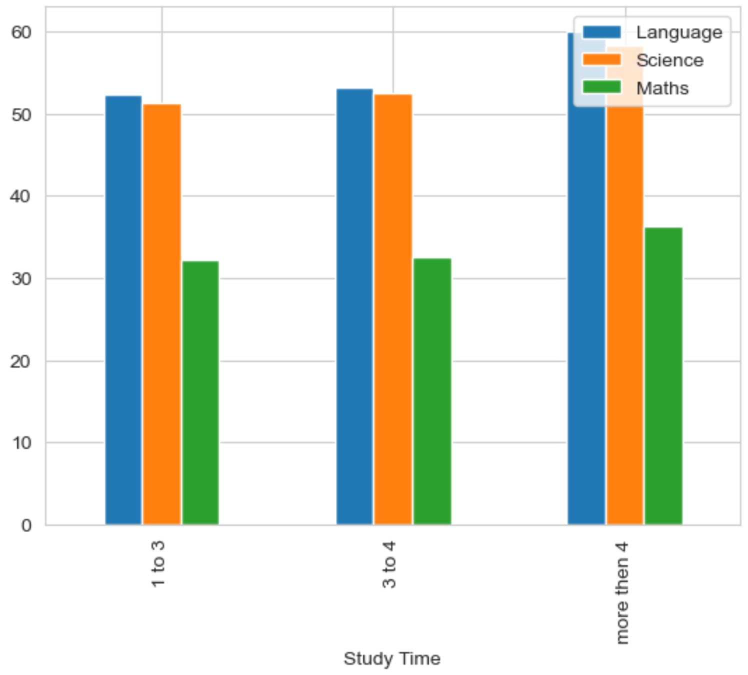
7. Effect of Study Time on Student's Performance

In [54]:

```
df.groupby("Study Time").agg({"Language" : "mean","Science" : "mean","Maths":"mean"}).plot(kind='bar')
```

Out[54]:

```
<Axes: xlabel='Study Time'>
```



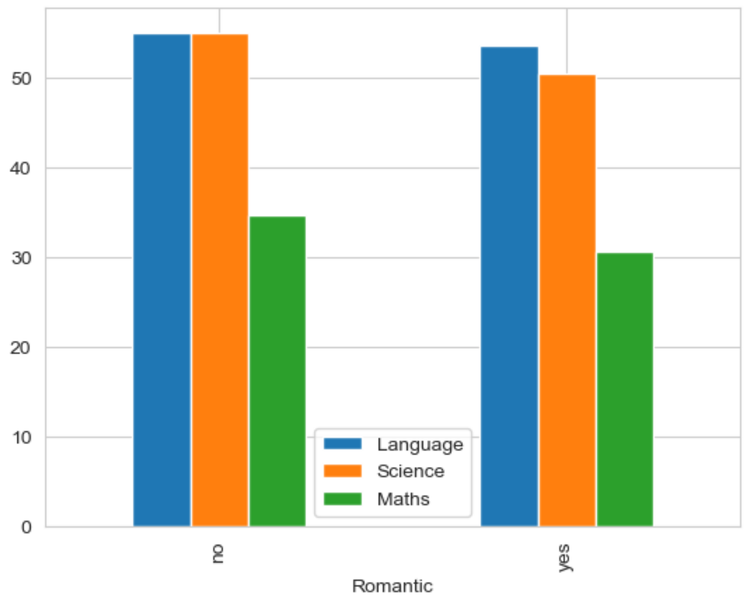
Conclusion

From the above analysis we can conclude that those student who spend their time in studing more then 4 hours has good performance in each subject and this is obvious that if you study more you will have better performance

8. Effect of Student's Relationship on their Performance

In [55]:
```
az = df.groupby("Romantic").agg({"Language" : "mean","Science" : "mean","Maths":"mean"})
az.plot.bar()
```

Out[55]:
```
<Axes: xlabel='Romantic'>
```



Conclusion

From the above analysis we can conclude that the students in relationship performed slightly less in compare to students not in relationship

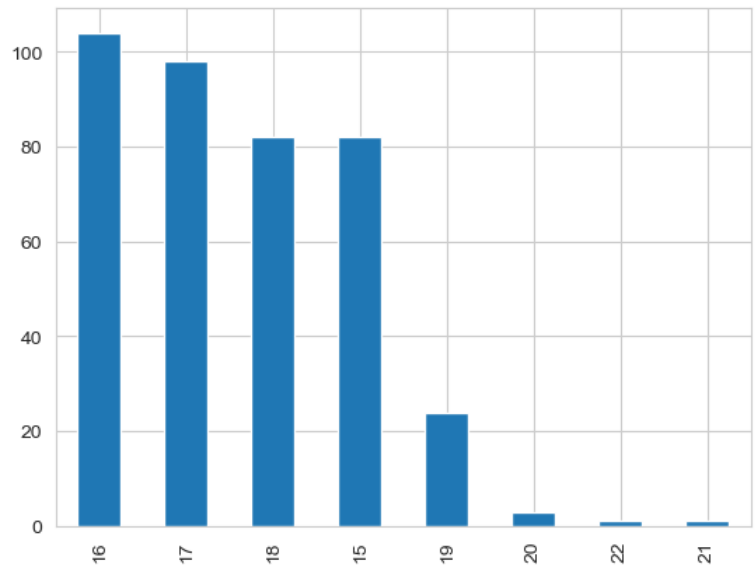
9. Comparision of different age groups in Schools (Counting)

In [56]:

```
df['Age'].value_counts().plot.bar()
```

Out[56]:

```
<Axes: >
```



Conclusion

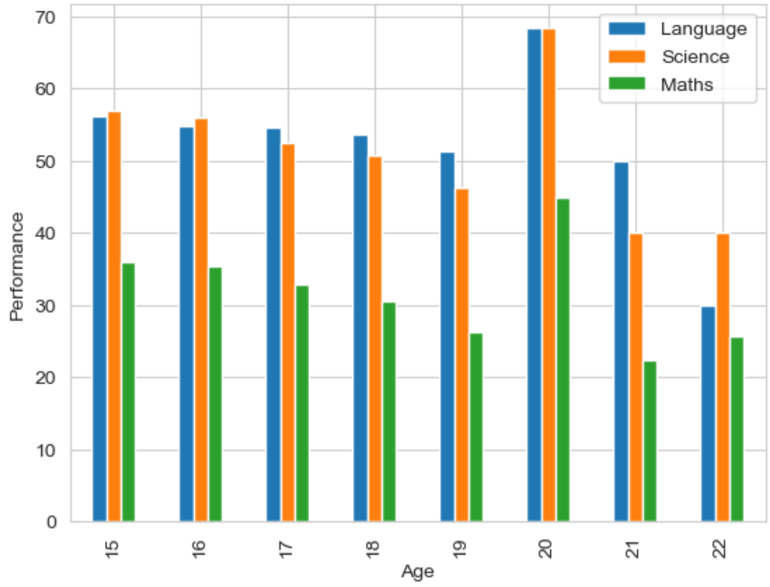
From the above analysis we can conclude that most of the students in both schools is between the age group 15-18 year

---

10. Comparision of Performance of different Age group

In [57]:

```
ageGroup = df.groupby("Age").agg({"Language" : "mean","Science" : "mean","Maths":"mean"})
ageGroup.plot(kind="bar", xlabel="Age",ylabel="Performance")
plt.show()
```



Conclusion

1. From the above analysis we can conclude that student of age 20 has much better performance in compare to other age group

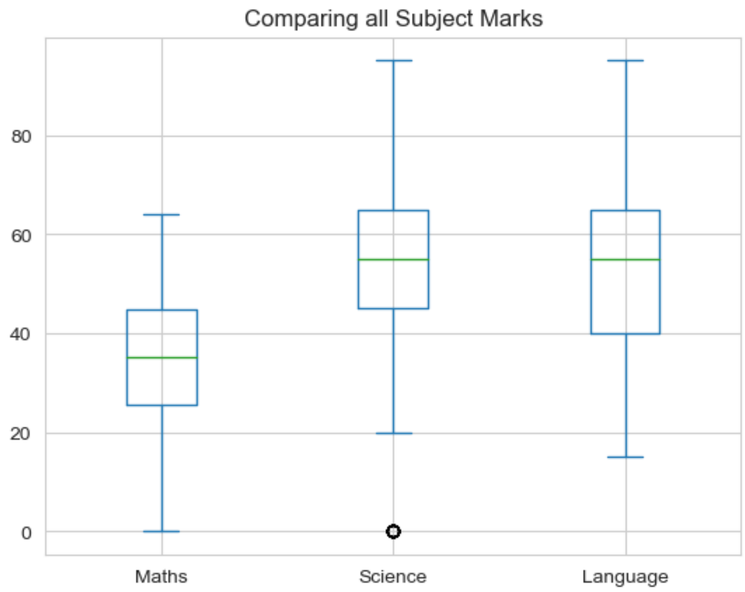
2. After age 20 the performance decreses constantly

---

11. Comparing Subjects (Language v/s Science v/s Maths)

In [58]:
```
df[['Maths','Science','Language']].plot(kind='box', title= "Comparing all Subject Marks")
```

Out[58]:
```
<Axes: title={'center': 'Comparing all Subject Marks'}>
```



Conclusion

Maths -

1. Students find maths difficult in compare to other subjects

2. Approx 75% Student in maths scores below 50 marks out of that 50% scores between 25 - 45 and othet 25% students scores below 25 Marks

3. Some students also scores 0 marks in Math

4. Highest Mark in Math is below 70

Science -

1. Each and every student scores above 20 marks in Science

2. Some Students also score above 80 in Science

3. 50% of Students scores between 45 to 65 in Science

Language -

1. Each and every student scores above 15 marks in Language

2. Some Students also score above 80 in Language

3. 50% of Students scores between 40 to 65 in Language

---

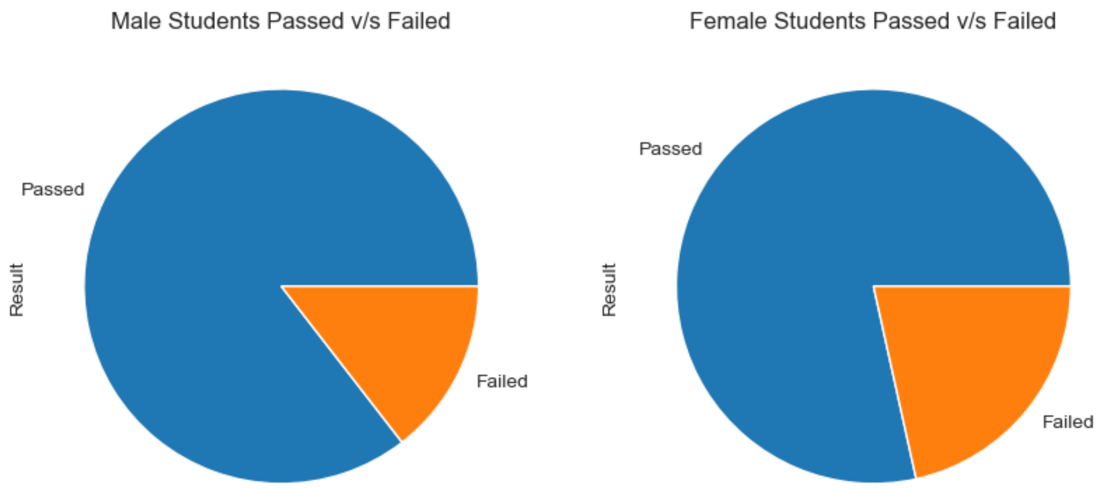
12. Comaprision of Passed v/s Failed Student

In [59]:
```
fig=plt.figure(figsize=(10,8), dpi=100)
df.loc[df['Percentage'] <= 33, 'Result'] = 'Failed'
df.loc[df['Percentage'] > 33, 'Result'] = 'Passed'

dmale = df.loc[df["Gender"] == "Male"]
dfemale = df.loc[df["Gender"] == "Female"]

plt.subplot(1,2,1)
dmale['Result'].value_counts().plot(kind='pie', title = "Male Students Passed v/s Failed")
plt.subplot(1,2,2)
dfemale['Result'].value_counts().plot(kind='pie' ,  title = "Female Students Passed v/s Failed")
```

Out[59]:
```
<Axes: title={'center': 'Female Students Passed v/s Failed'}, ylabel='Result'>
```



Conclusion

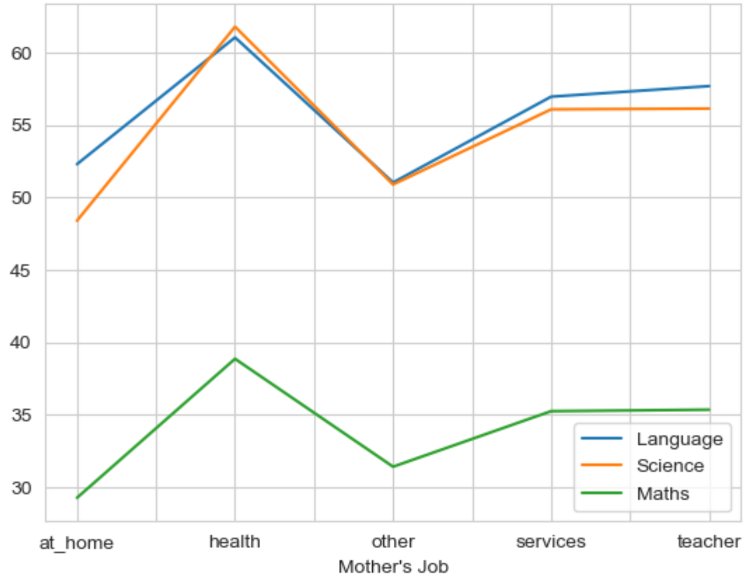
From the above analysis we can conclude that the passing percentage in male student is more in compare to female Students

---

13. Effect of Mother's Job on Student's Academic Performace

```
In [60]: mj = df.groupby("Mother's Job").agg({"Language" : "mean","Science" : "mean","Maths":"mean"})
         mj.plot(kind="line")
```

```
Out[60]: <Axes: xlabel="Mother's Job">
```



Conclusion

1. Those students whose mother is working in health sector have good performance in all subjects

2. Those students whose mother is a teacher also performed well.

3. Those students whose mother is working in Health Sector scores highest in Science in compare to Maths and Language

3. Those students whose mother is a teacher scores highest in Language in compare to Maths and Science

---

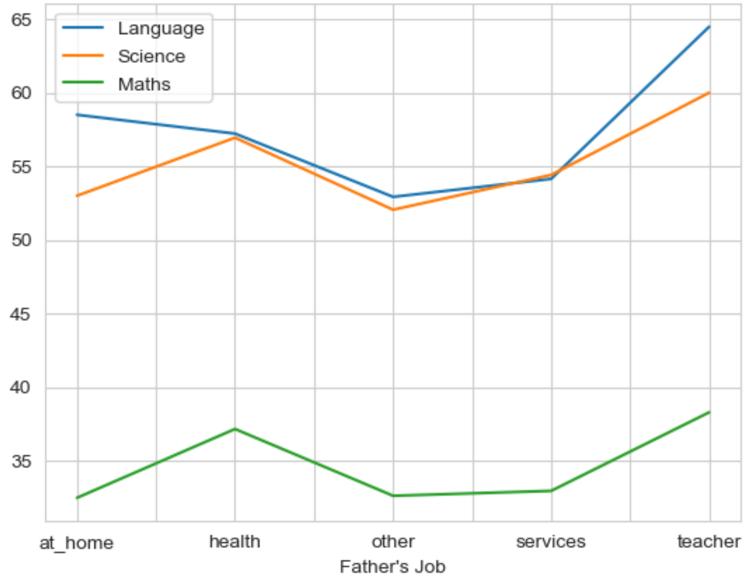
14. Effect of Father's Job on Student's Academic Performace

In [61]:

```
fj = df.groupby("Father's Job").agg({"Language" : "mean","Science" : "mean","Maths":"mean"})
fj.plot(kind="line")
```

Out[61]:

```
<Axes: xlabel="Father's Job">
```



Conclusion

1. Those students whose father is a teacher have good performance in all subjects

2. Those students whose father is working in health sector also performed well.

3. Those students whose father is working in Health Sector scores highest in Science in compare to Maths and Language

3. Those students whose father is a teacher scores highest in Language in compare to Maths and Science

---

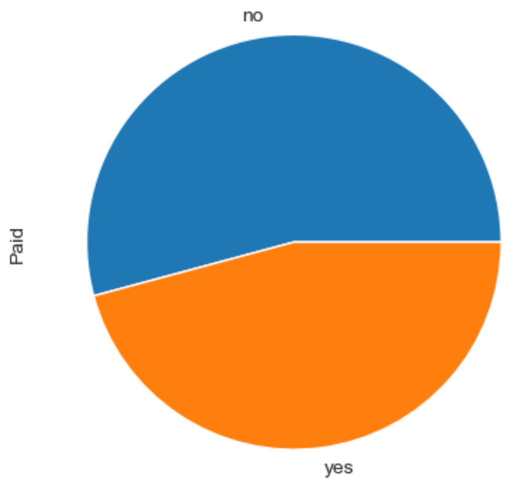
15. Comaprision of Students who take Extra Paid Classes (Counting)

In [62]:

```
df["Paid"].value_counts().plot(kind="pie")
```

Out[62]:

```
<Axes: ylabel='Paid'>
```



Conclusion

From the above analysis we can conclude that most of the students (Approx 40% students) attend extra paid classed

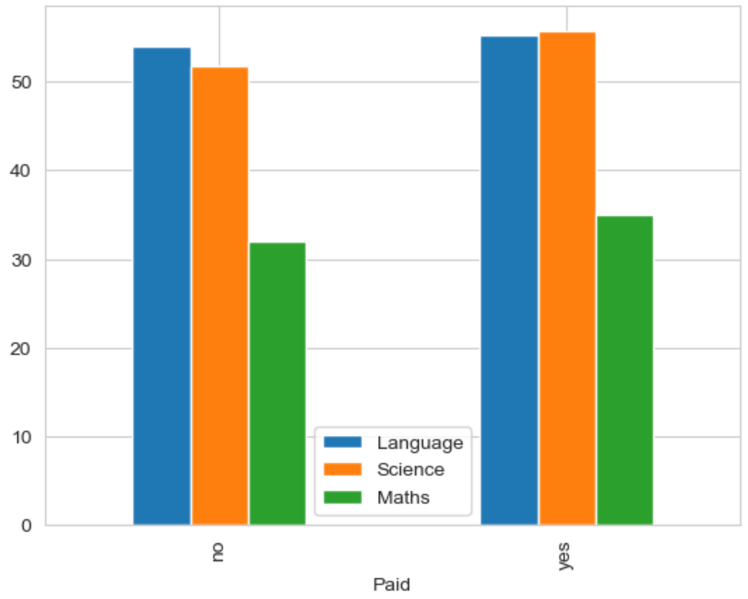
16. Comaprision of Students who take Extra Paid Classes (Performance)

In [63]:

```
ispaid = df.groupby("Paid").agg({"Language" : "mean","Science" : "mean","Maths":"mean"})
ispaid.plot(kind="bar")
```

Out[63]:

```
<Axes: xlabel='Paid'>
```



Conclusion

From the above analysis we can conclude that those students who attended extra paid classes performed slightly better in compare to other students

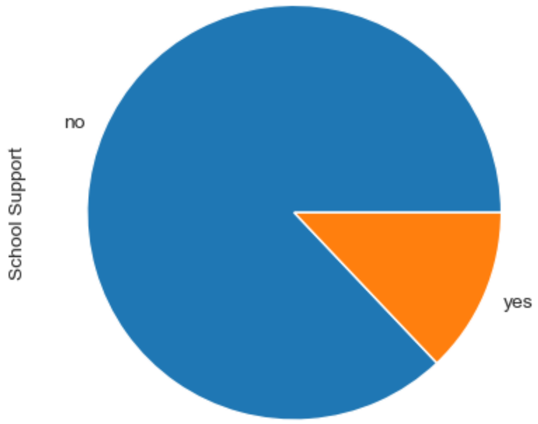
17. Comaprision of Students who has School Support (Counting)

In [64]:

```
df["School Support"].value_counts().plot(kind="pie")
```

Out[64]:

```
<Axes: ylabel='School Support'>
```



Conclusion

From the above analysis we can conclude that very few student has school support

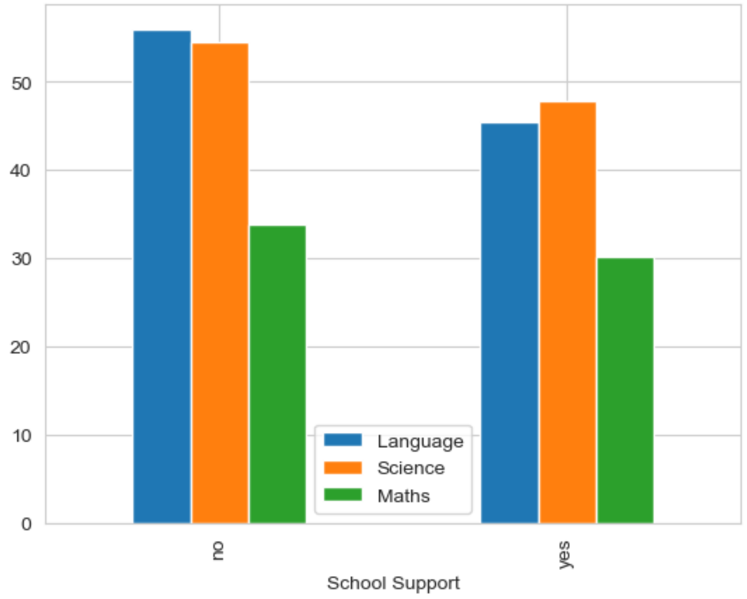
18. Comaprision of Students who has School Support (Performance)

In [65]:

```
isscsupp = df.groupby("School Support").agg({"Language" : "mean","Science" : "mean","Maths":"mean"})
isscsupp.plot(kind="bar")
```

Out[65]:

```
<Axes: xlabel='School Support'>
```



Conclusion

From the above analysis we can conclude that those students who has school support liitle bit weaker in performance

---

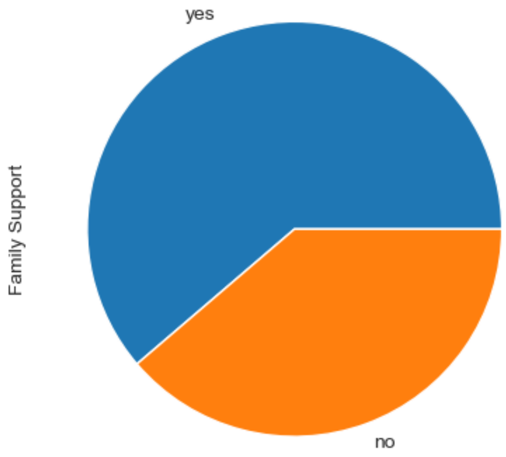
19. Comaprision of Students who has Family Support (Counting)

In [66]:

```
df["Family Support"].value_counts().plot(kind="pie")
```

Out[66]:

```
<Axes: ylabel='Family Support'>
```



Conclusion

From the above analysis we can conclude that most of the students has family support but still there are more then 30% students who has no family support

---

20. Comaprision of Students who Participate in Extra curicullar Activity

In [67]:

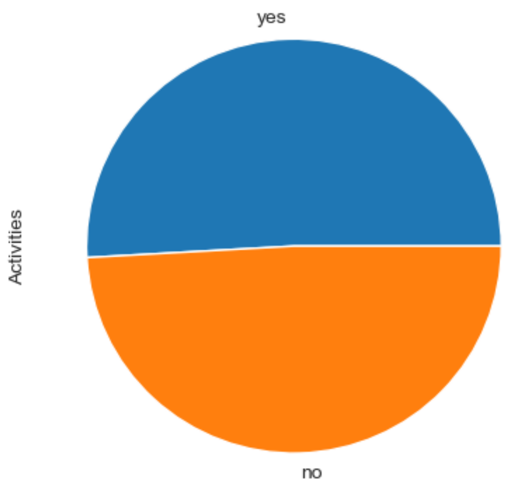
```
df["Activities"].value_counts().plot(

    kind="pie"

)
```

Out[67]:

```
<Axes: ylabel='Activities'>
```



Conclusion

From the above analysis we can conclude that approx 50% of students do not participate in any extra-curricular activity which is quite disappointing

---

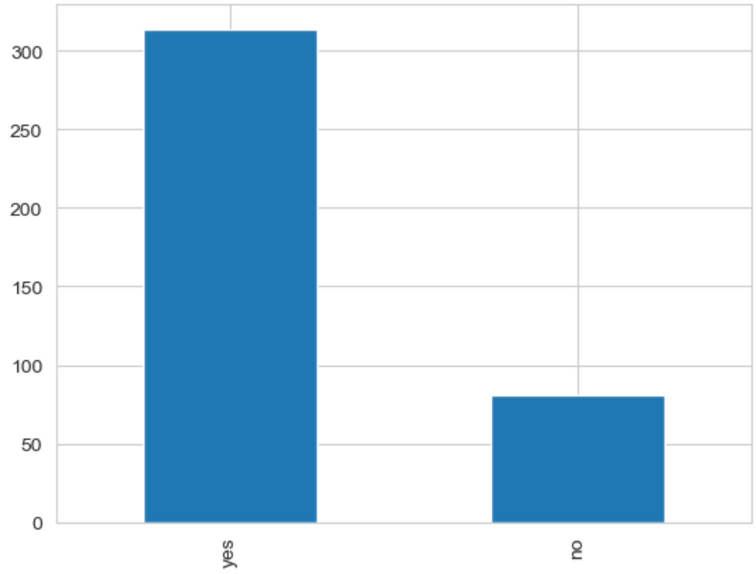
21. Comaprision of Students who Attend their Nursery in childhood

In [68]:

```
df["Nursery"].value_counts().plot(kind="bar")
```

Out[68]:

```
<Axes: >
```



22. Comaprision of Students who has Internet Access (Counting)

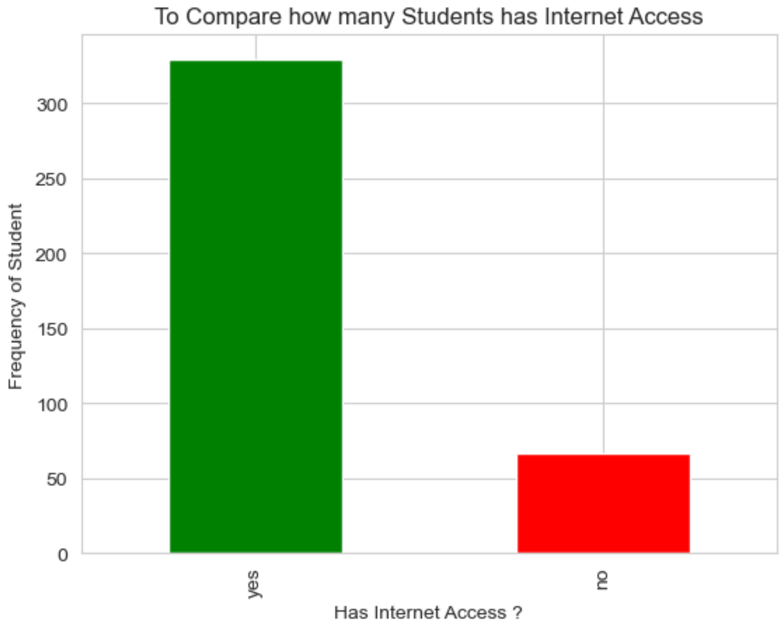
In [69]:

```
df["Internet"].value_counts().plot(

    kind="bar",
    color = ['green', 'red'],
    xlabel="Has Internet Access ?",
    ylabel="Frequency of Student",
    title="To Compare how many Students has Internet Access"

)
```

```
Out[69]: <Axes: title={'center': 'To Compare how many Students has Internet Access'}, xlabel='Has Internet Access ?', ylabel='Frequency
         of Student'>
```



Conclusion

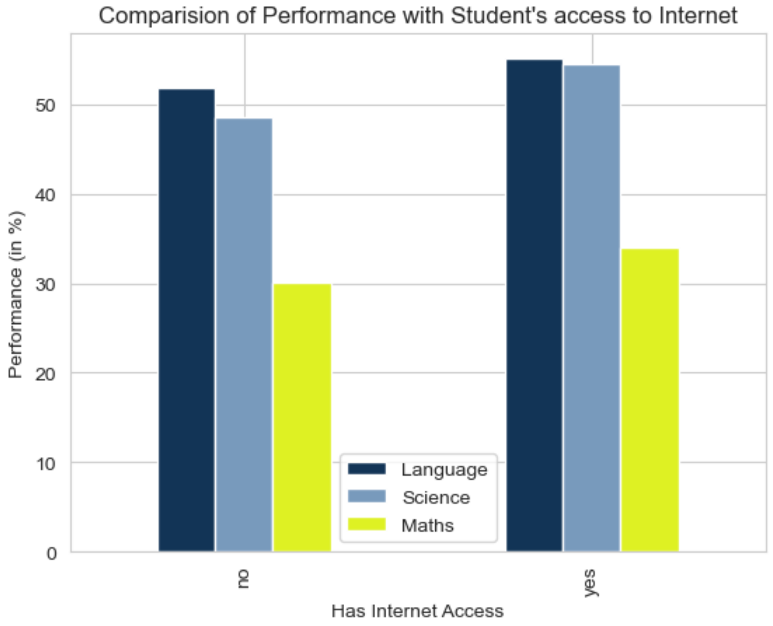
From the above analysis we can conclude that nearly 55 students are there who have not attended their nursery school

---

23. Comaprision of Students who has Internet Access (Performance)

```
In [70]: colors = ['#123456', '#789ABC', '#DEF123']
         nur = df.groupby("Internet").agg({"Language" : "mean","Science" : "mean","Maths":"mean"})
         nur.plot(
             kind="bar",
             color=colors,
             xlabel="Has Internet Access",
             ylabel="Performance (in %)",
             title="Comparision of Performance with Student's access to Internet"
         )
```

```
Out[70]: <Axes: title={'center': "Comparision of Performance with Student's access to Internet"}, xlabel='Has Internet Access', ylabel
         ='Performance (in %)'>
```



Conclusion

From the above analysis we can conclude that student have internet access performed slightly better then who don't have internet access

---

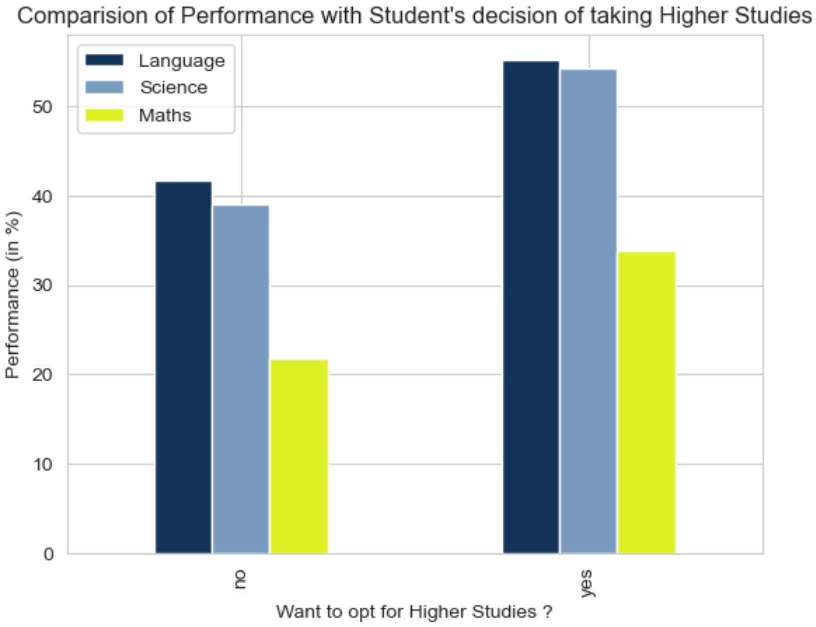
24. Comaprision of Students who want to opt for Higher Studies

In [71]:

```
colors = ['#123456', '#789ABC', '#DEF123']
nur = df.groupby("Higher").agg({"Language" : "mean","Science" : "mean","Maths":"mean"})
nur.plot(
    kind="bar",
    color=colors,
    xlabel="Want to opt for Higher Studies ?",
    ylabel="Performance (in %)",
    title="Comparision of Performance with Student's decision of taking Higher Studies"
)
```

Out[71]:

```
<Axes: title={'center': "Comparision of Performance with Student's decision of taking Higher Studies"}, xlabel='Want to opt fo
r Higher Studies ?', ylabel='Performance (in %)'>
```



Conclusion

From the above analysis we can conclude that those students who performed well wants to take higher education and those who didn't performed well don't want to take higher education

---

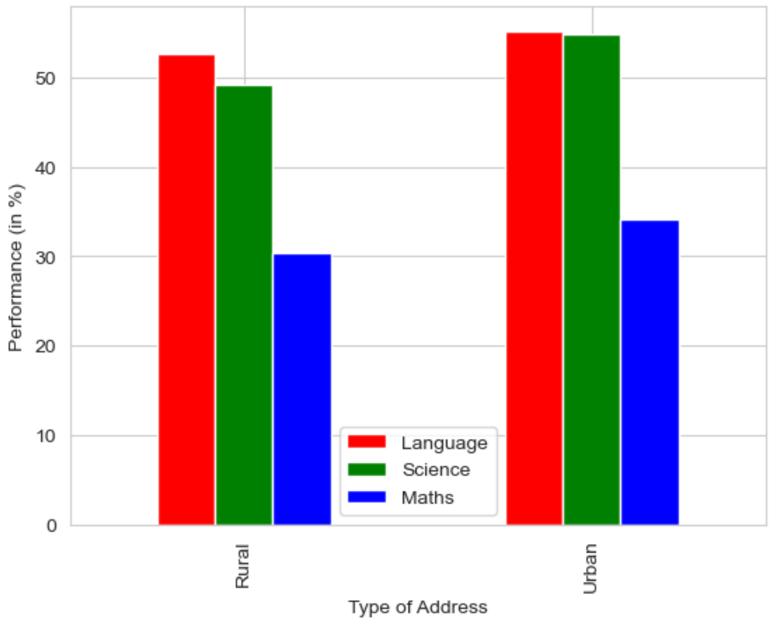
25. Effect of Address type on Student's Academic Performance

In [72]:

```
colors = ['red', 'green', 'blue']
df.groupby("Address").agg(
    {"Language" : "mean","Science" : "mean","Maths":"mean"}
).plot(kind="bar",xlabel="Type of Address", ylabel="Performance (in %)",color=colors)
```

Out[72]:

```
<Axes: xlabel='Type of Address', ylabel='Performance (in %)'>
```



Conclusion

From the above analysis we can conclude that there is very few effect of address type on student's academic performace

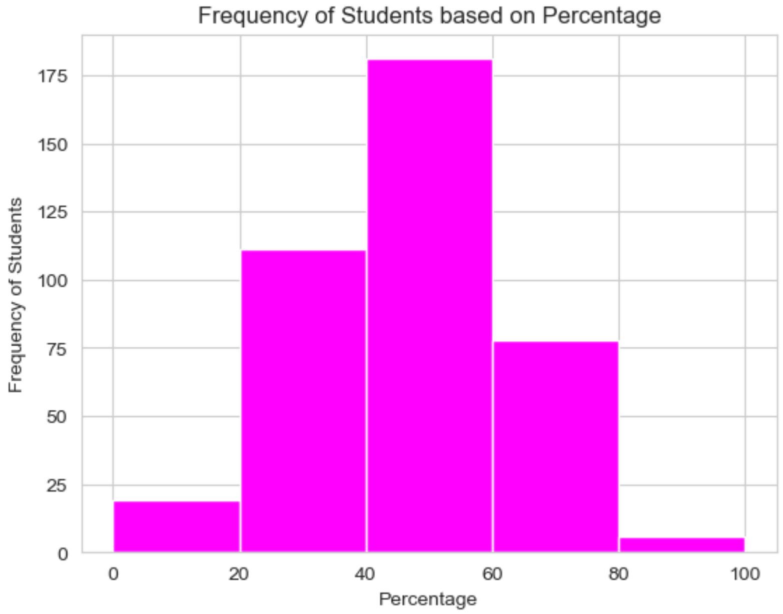
26. Percentage Scored v/s Frequency of Student

In [73]:

```
fig, ax = plt.subplots(1, 1)
ax.hist(df['Percentage'], bins = [0, 20, 40, 60, 80,100],color="magenta")
ax.set_title("Frequency of Students based on Percentage")
ax.set_xlabel('Percentage')
ax.set_ylabel('Frequency of Students')
```

Out[73]:

```
Text(0, 0.5, 'Frequency of Students')
```



Conclusion

From the above analysis we can conclude that most of the student scores between 40-60% and very few student was able to score 80% above

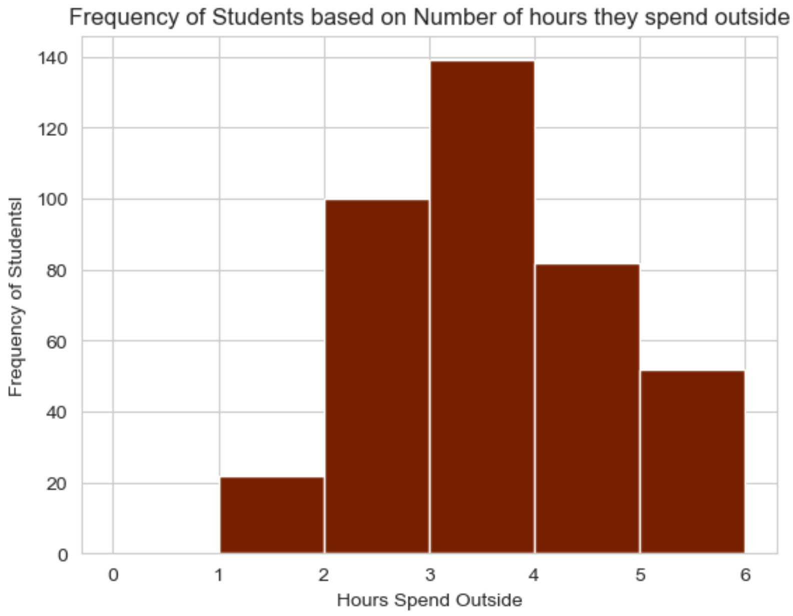
27. Comparision on number of hours student spend outside their home and their frequency

In [74]:

```
fig, ax = plt.subplots(1, 1)
ax.hist(df['GoOut'], bins = [0,1,2,3,4,5,6],color="#772000")
ax.set_title("Frequency of Students based on Number of hours they spend outside")
ax.set_xlabel('Hours Spend Outside')
ax.set_ylabel('Frequency of Studentsl')
```

Out[74]:

```
Text(0, 0.5, 'Frequency of Studentsl')
```



Conclusion

From the above analysis we can conclude that more then 120 student go out for 3-4 hours a day

---

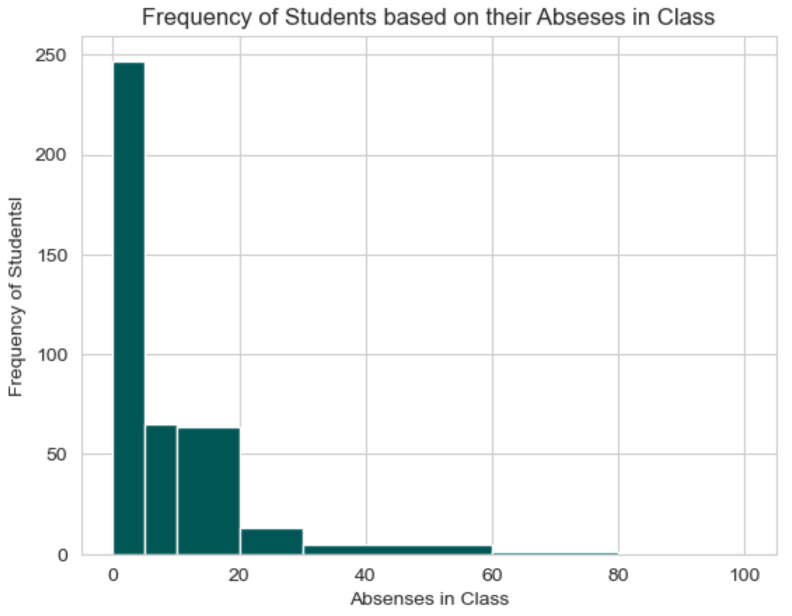
28. Absenses in Class v/s Number of Students

In [75]:

```
fig, ax = plt.subplots(1, 1)
ax.hist(df['Absences'], bins = [0,5,10,20,30,60,80,100],color="#005555")
ax.set_title("Frequency of Students based on their Abseses in Class")
ax.set_xlabel('Absenses in Class')
ax.set_ylabel('Frequency of Studentsl')
```

Out[75]:

```
Text(0, 0.5, 'Frequency of Studentsl')
```



Conclusion

From the above analysis we can conclude that about 250 students has absenses between 0-5 days and there are very few student who missed their class more then 20 days

---

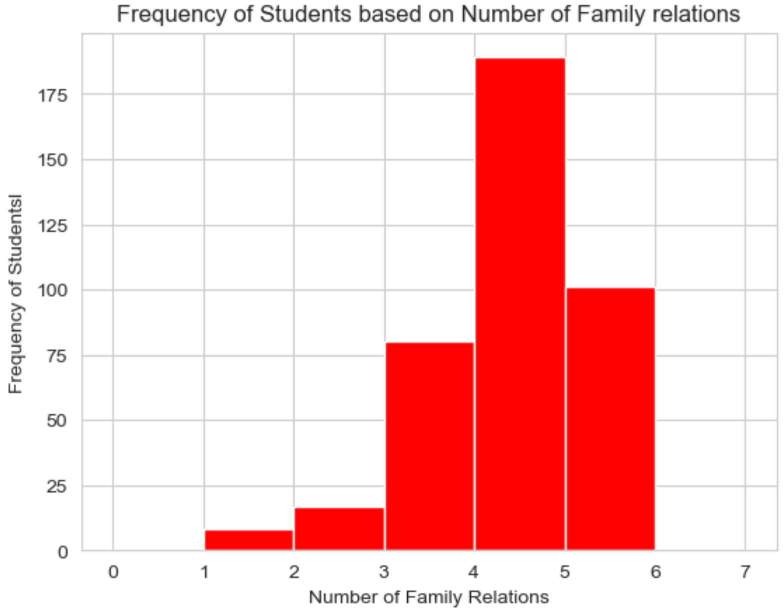
29. Family Relation v/s Number of Students

In [76]:

```
fig, ax = plt.subplots(1, 1)
ax.hist(df['FamRel'], bins = [0,1,2,3,4,5,6,7],color="red")
ax.set_title("Frequency of Students based on Number of Family relations")
ax.set_xlabel('Number of Family Relations')
ax.set_ylabel('Frequency of Studentsl')
```

Out[76]:

```
Text(0, 0.5, 'Frequency of Studentsl')
```



Conclusion

From the above analysis we can conclude that most of the student having their family between 4-5 Members

---

Summary

- From above analysis we can conclude that the number of Female Students is more then the number of Male Students
- From above analysis we can conclude that the number of Female Students is more then the number of Male Students in both the Schools
- From above analysis we can conclude that male students performed little good in compare to Female Students
- From the above analysis we can conclude that those students who has more family member is little bit distracted and has low performance in compare to those students who has comparatively less family size
- From the above analysis we can conclude that the Mother's Education has good impact on child's academic Performace
- From the above analysis we can conclude that the Father's Education has good impact on child's academic Performace
- From the above analysis we can conclude that those student who spend their time in studing more then 4 hours has good performance in each subject and this is obvious that if you study more you will have better performance
- From the above analysis we can conclude that those student who spend their time in studing more then 4 hours has good performance in each subject and this is obvious that if you study more you will have better performance
- From the above analysis we can conclude that most of the students in both schools is between the age group 15-18 year
- From the above analysis we can conclude that student of age 20 has much better performance in compare to other age group
- After age 20 the performance decreses constantly
- Students find maths difficult in compare to other subjects
- Approx 75% Student in maths scores below 50 marks out of that 50% scores between 25 - 45 and othet 25% students scores below 25 Marks
- Some students also scores 0 marks in Math
- Highest Mark in Math is below 70
- Each and every student scores above 20 marks in Science

- Some Students also score above 80 in Science
- 50% of Students scores between 45 to 65 in Science
- Each and every student scores above 15 marks in Language
- Some Students also score above 80 in Language
- 50% of Students scores between 40 to 65 in Language¶
- From the above analysis we can conclude that the passing percentage in male student is more in compare to female Students
- Those students whose mother is working in health sector have good performance in all subjects
- Those students whose mother is a teacher also performed well.
- Those students whose mother is working in Health Sector scores highest in Science in compare to Maths and Language
- Those students whose mother is a teacher scores highest in Language in compare to Maths and Science
- Those students whose father is a teacher have good performance in all subjects
- Those students whose father is working in health sector also performed well.
- Those students whose father is working in Health Sector scores highest in Science in compare to Maths and Language
- Those students whose father is a teacher scores highest in Language in compare to Maths and Science
- Those students whose father is a teacher have good performance in all subjects
- Those students whose father is working in health sector also performed well.
- Those students whose father is working in Health Sector scores highest in Science in compare to Maths and Language
- Those students whose father is a teacher scores highest in Language in compare to Maths and Science
- From the above analysis we can conclude that those students who attended extra paid classes performed slightly better in compare to other students
- From the above analysis we can conclude that very few student has school support
- From the above analysis we can conclude that those students who has school support liitle bit weaker in performance
- From the above analysis we can conclude that most of the students has family support but still there are more then 30% students who has no family support
- From the above analysis we can conclude that approx 50% of students do not participate in any extra-curricular activity which is quite disappointing
- From the above analysis we can conclude that nearly 55 students are there who have not attended their nursery school
- From the above analysis we can conclude that student have internet access performed slightly better then who don't have internet access
- From the above analysis we can conclude that those students who performed well wants to take higher education and those who didn't performed well don't want to take higher education
- From the above analysis we can conclude that there is very few effect of address type on student's academic performace
- From the above analysis we can conclude that most of the student scores between 40-60% and very few student was able to score 80% above
- From the above analysis we can conclude that more then 120 student go out for 3-4 hours a day
- From the above analysis we can conclude that about 250 students has absenses between 0-5 days and there are very few student who missed their class more then 20 days
- From the above analysis we can conclude that most of the student having their family between 4-5 Members