In [ ]:
```
# Simple Linear Regression Chapter

Import the Libraries:

1. Numpy - Math functions
2. MatplotLib - Visualization
3. SciKitLearn
    a) Build Models
    b) Preprocessing (Encoding,One-Hot Encoding)

4. Pandas Package
    a) Read Files into Dataframes
    b) Create, Manipulate Dataframes
    c) Create Feature Matrix
    d) Create Output Vector

SciKit Learn Details:

Preprocess Data
a) Imputer - Missing values using strategies like Mean
b) Convert Categorical data to Numbers - LabelEncoder
c) Convert Numbers into One Hot Encoding - OneHotEncoder
d) Model Selection - Split into Training and Test Set Data
e) Scaling Data - Standaradization / Normalization

Build Simple Linear Regression Models
a)  linear_model library, LinearModel object
```

Pre-processing for Simple Linear Regression Model

In [5]:
```
# Simple Linear Regression

#-----------------Preprocessing Section ------------------------------
-----
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Salary_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/
3, random_state = 0)

#---------------End of Pre-processing Section -------------------------
-------
```

Fitting the Model (to Training set) and Predicting (on Test set)

Use the LinearRegression Object from the linear_model library Use the Fit method of the Regressor object to the training Set Use the Predict methd of the Regressor object on the Test Set

In [7]:
```
# Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)
```

Visualizing the Training Set

Use a scatter plot for the training set Use a Line to predict the training inputs, and predicted outputs on training set

In [10]:
```
# Visualising the Training set results
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

Visualizing the Test Set

Use a scatter plot for the test set Use a Line to predict the training inputs, and predicted outputs on training set

In [11]:
```
# Visualising the Test set results
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```