# Machine Learning – WS'12
# Exercise-3

**Prof. Dr. Dr. Lars Schmidt-Thieme, Umer Khan**
**Information Systems and Machine Learning Lab (ISMLL),**
**University of Hildesheim**

## Problem-1:

Implement linear regression with multiple variables to predict the prices of houses. The housing.txt contains a training set of housing prices in Hildesheim. The first column is the size of the house (in square feet), the second column is the number of bedrooms, and the third column is the price of the house. In the script, multi.m, you should start with loading data as mentioned in the comments. The script also has code to print first few examples. By looking at the values, note that house sizes are about 1000 times the number of bedrooms. When features differ by orders of magnitude, first performing feature scaling can make gradient descent converge much more quickly.

### Normalize Features

So first step is to implement 'featureNormalize.m'. As we discussed in class, subtract each value of each feature with the mean of this feature. Then divide the feature values with standard deviation of this feature. In Octave, you can calculate standard deviation with 'std' function like std('X(:,1)') gives you standard deviation of first feature in matrix X.

### Computing Cost Function and Gradient Descent for Multiple Variables

In Exercise-2, you calculated Gradient Descent for single variable linear regression. The only difference is that now you have to modify your code so that it works for any number of features. You already know the 'size' command to find number of features in X. The hypothesis function and the batch gradient descent update rule remain unchanged.

Implement computeCostMulti.m and gradientDescentMulti.m.

Using script in multi.m, plot the cost function 'J' computed in each iteration against the number of Iterations. Using this plot, you can analyze whether chosen value of alpha works well or you need a change for it. Try different values of alpha (learning rate) for learning gradient descent.