# Week9 Notes

### Commercial Solvers

- Popular Commercial Solvers:
  - CPLEX - IBM: https://www.ibm.com/products/ilog-cplex-optimization-studio
  - Gurobi: https://www.gurobi.com/
  - FICO - Xpress: https://www.fico.com/en/products/fico-xpress-optimization
- We will use Gurobi for this course.
- Gurobi Installation:
  - Register: https://pages.gurobi.com/registration Register with your .ozu e-mail to get Academic Licence
  - Download page: https://www.gurobi.com/downloads/
  - Default download directory for macOS: /Library/gurobi
  - Default download directory for Windows:  C:\gurobi\
  - When you get the academic licence, you will see <grbgetkey “numbers”>. Copy and paste it to the terminal/cmd.
  - Terminal/Cmd will ask you to where to locate Gurobi licence, give the download directory.
  - After that, should add .jar to the path to use Gurobi.
  - If you have any problem, please contact with TA’s.

### Introduction to Gurobi

- Interface - Inheritance: https://www.geeksforgeeks.org/interfaces-and-inheritance-in-java/
- javac -main_method-.java
- java -main_method-
- How to work with different jars? >> JAR: Java ARchive
  - javac -clashpath <path_to_jar>:. -main_method-.java
  - java -clashpath <path_to_jar>:. -main_method-
- Also you can add external jar with IDE.

### Solving Knapsack Problem with Gurobi

- import gurobi.* >> Import all Gurobi files
- GRBEnv -env_name- = new GRBEnv (“-name-.log”);  >> Creating Gurobi environment
- GRBModel -model_name- = new GRBModel(-env_name-); >> Creating Gurobi model
- GRBVar[ ] -var_name- = new GRBVar[-size-] >> Creating GRBVar array (Decision Variables)
- -var_name-[-index-] = -model_name-.addVar(-lower_bound_value-, -upper_bound_value-, -objective_value-, -type_of_variable-, -name_of_variable-); >>

Add variable to the model
    - Type of variable can be GRB.BINARY, GRB.INTEGER, GRB.CONTINUOUS etc. For more details, check Gurobi reference.
- GRBLinExpr -expression_name- = new GRBLinExpr(); >> Creating new Linear Expression
- -expression_name-.addTerm(-coefficient_value-, -gurobi_variable-); >> Add -coefficient_value- * -gurobi_variable- to the expression
- -model_name-.addConstr( -expression_name- , -inequality- , -right_hand_side- , -name_of_the_constraint-); >> Add constraint to the model
    - Inequalities > GRB.LESS_EQUAL , GRB.GREATER_EQUAL, GRB.EQUAL etc. For more details, check Gurobi reference.
- -model_name-.setObjective( -expression_name-, -objective_preference-); >> Add objective to the model
    - Objective Preference > GRB.MINIMIZE, GRB.MAXIMIZE
- -model_name-.write("-file_name-.lp"); >> Writes model to the -file_name-.lp file
- -model_name-.optimize() >> Optimize model
- -gurobi_variable-.get(GRB.DoubleAttr.X) >> Get optimal value of Gurobi variable, after optimization.
- -model_name-.dispose(); >> Clear model
- -environment_name-.dispose(); >> Clear environment
- There are lots of different features of Gurobi. For more information and details, you can check Documentation: https://www.gurobi.com/documentation/