

# Learning MP for ILOG OPL Development Studio

## Courses

This is the first in a training program in 2 courses, designed to show analysts, designers and programmers how to use an integrated development environment in optimization projects:

- Mathematical Programming key concepts (for non-OR Specialists) are explored in a 1-day course, *Learning MP for OPL Development Studio*:
- Hands-on practice building OPL programs, accompanied by explanatory lectures and demonstrations of the OPL language, form the core of the 3-day course (for everyone), *Optimization with ILOG OPL Development Studio*:

## Who should take this course

Analysts, IT experts, programmers, decision makers and others who need to build optimization models using OPL and are not experts in Operations Research (OR).

## Outline

### *MP and you*

About you  
About MP  
OPL Development Studio  
MP terminology  
Operations Research  
More about optimization

### *Linear Programming*

About linear programming  
Example: a production process problem  
Infeasible models  
Unboundedness  
Algorithms for solving LP problems

## Objectives

Acquire basic knowledge about Mathematical Programming (MP) techniques and terminology, so as to be able to follow the *Optimization with OPL Development Studio* course

## Prerequisites

- No course prerequisites
- This course is a prerequisite for *Optimization with OPL Development Studio* for trainees who do not have experience in Operations Research (OR)

## Length

- *Learning MP for OPL Development Studio*: 1 day
- *Optimization with OPL Development Studio*: 3 days
- 9:00 am – 6:00 pm

---

## *Beyond "Simple" LP*

Network models  
Nonlinear models and convexity  
When the problem isn't neat – piecewise linear programming  
Working with integers  
About quadratic programming

## *Modeling Practice*

Models and the real world  
Presolve  
The importance of sparsity  
A packing model for "what-if" analysis  
Tips for better models

An appendix is provided with a review of linear algebra and details about the simplex algorithm