Timetable Problem

In [27]:
```
import sys
import pandas as pd
import numpy as np

from gamspy import (
    Container,Set,Alias,Parameter,Variable,Equation,Model,Problem,Sense,Opti
    Domain,Number,Sum,Product,Smax,Smin,Ord,Card,SpecialValues,
    ModelStatus,SolveStatus,
)

options = Options(time_limit=10000000,relative_optimality_gap=0.)

m = Container(options=options)
```

In [28]:
```
t = Set(m,'t',description='TEACHERS',
    records = [
        ('MrCheese', 'English'), ('MrsInsulin', 'Biology'),
        ('MrMap', 'History-Geography'), ('MrEffofecks', 'Mathematics'),
        ('MrsDerivate', 'Mathematics'), ('MrsElectron', 'Physics'),
        ('MrWise', 'Philosophy'), ('MrMuscle', 'Sport'),
        ('MrsBiceps', 'Sport')])
c = Set(m,'c',description='CLASS',records=[1,2])
d = Set(m,'d',description='DAYS',records=["Mon","Tue","Wed","Thu","Fri"])
# Number of time periods for courses
NP=4
l = Set(m,'l',description='time slots for the entire week',
    records=[ind+1 for ind in range(NP)])

COURSE = Parameter(m,domain=[t,c],records=np.array([
    [1, 1],[3, 3],[2, 2],[0, 4],[4, 0],[3, 3],[1, 1],[1, 0],[0, 1]]))

# Input your model here

x = m.addVariable('x', 'binary', domain=[t, c, d, l])

# Equations:

# 1. Each class must be planned
plan_courses = Equation(m, 'course_req', domain=[t, c])
plan_courses[t, c] = Sum((d, l), x[t, c, d, l]) == COURSE[t, c]

# 2. Each class must have exactly one lesson per day
assign_class = m.addEquation('assign_class', domain=[c, d, l])
assign_class[c, d, l] = Sum(t, x[t, c, d, l]) <= 1

# 3. Each teacher can only teach one class at a time
assign_teacher = m.addEquation('assign_teacher', domain=[t, l, d])
assign_teacher[t, l, d] = Sum(c, x[t, c, d, l]) <= 1

# 4. Every subject only once per day
```

```
# 5. Sport must be on Thursday afternoon (slot 3)
sport_constraint = m.addEquation('sport_constraint')
sport_constraint[:] = x["MrsBiceps", "2", "Thu", "3"] + x["MrMuscle", "1", "

# 6. No lessons on Monday morning (slot 1)
no_classes_monday_morning = m.addEquation('no_classes_monday_morning', domai
no_classes_monday_morning[t, c] = x[t, c, "Mon", "1"] == 0

# 7. Mr. Effofecks is absent Monday morning (slot 1)
mr_effofecks_absent = m.addEquation('mr_effofecks_absent', domain=[c])
mr_effofecks_absent[c] = x["MrEffofecks", c, "Mon", "1"] + x["MrEffofecks",

# 8. Mrs. Insulin does not work on Wednesday
mrs_insulin_absent = m.addEquation('mrs_insulin_absent', domain=[c, l])
mrs_insulin_absent[c, l] = x["MrsInsulin", c, "Wed", l] == 0


timetab = m.addModel('dragon',
    equations=m.getEquations(),
    problem=Problem.MIP,
    sense=Sense.MIN,
    objective=Sum((t, c, d), x[t, c, d, "1"] + x[t, c, d, "4"]),
)

timetab.solve()
```

Out[28]:

	Solver Status	Model Status	Objective	Num of Equations	Num of Variables	Model Type	Solver	Solve Tim
0	Normal	OptimalGlobal		10	268	361	MIP	CPLEX	0.0

In [29]:

```
lessons = Set(m,'lessons',[t,c,d,l])
lessons[t,c,d,l] = x.l[t, c, d, l] == 1
display(lessons.pivot(fill_value=''))
```

			1	2	3	4
MrCheese	**1**	**Mon**		True		
	2	**Mon**			True	
MrsInsulin	**1**	**Mon**			True	
		Fri		True	True	
	2	**Mon**		True		
		Tue		True	True	
MrMap	**1**	**Tue**		True	True	
	2	**Fri**		True	True	
MrEffofecks	**2**	**Wed**	True	True	True	
		Thu		True		
MrsDerivate	**1**	**Tue**				True
		Wed	True	True	True	
MrsElectron	**1**	**Wed**				True
		Thu	True	True		
	2	**Thu**			True	True
		Fri	True			
MrWise	**1**	**Thu**				True
	2	**Wed**				True
MrMuscle	**1**	**Thu**			True	
MrsBiceps	**2**	**Thu**	True			