A Production Planning Problem

```
# model : Production Planning Problem
m = gp.Model("PPP")

# create decision variables
r1 = m.addVar(name ="1_regular")
r2 = m.addVar(name ="2_regular")
r3 = m.addVar(name ="3_regular")
r4 = m.addVar(name ="4_regular")
ep1 = m.addVar(name ="1_extra")
ep2 = m.addVar(name ="2_extra")
ep3 = m.addVar(name ="3_extra")
ep4 = m.addVar(name ="4_extra")

# total production
prod_m1 = r1+ep1
prod_m2 = r2+ep2
prod_m3 = r3+ep3
prod_m4 = r4+ep4

# Storage
    #stor_m1 = prod_m1 - 120
    #stor_m2 = prod_m2 - 160
    #stor_m3 = prod_m3 - 300
    #stor_m4 = prod_m4 - 200
stor_m1 = prod_m1-120
stor_m2 = prod_m2+(stor_m1)-160
stor_m3 = prod_m3+(stor_m2)-300
stor_m4 = prod_m4+(stor_m3)-200

# Storage total cost
# sum ( each month storage * $5)
stor_total_cost = (stor_m1 * 5) + (stor_m2 * 5) + (stor_m3 * 5) + (stor_m4 * 5)

# Regular production total cost
# sum( each month regular production * each moth production cost)
# sum( regular procuction * ci )
rp_total_cost = (r1+r2+r3+r4) * 440

# Extra production total cost
# sum(Eeach month extra production * each month extra production cost)
ep_total_cost = (ep1+ep2+ep3+ep4) * (440 + 260)

# the objective : minimize the cost
    #storage --> previous month...
# sum(each month storage * 5 dollar) + sum(regular proudction total cost) + sum(extra production cost)

obj = stor_total_cost + rp_total_cost + ep_total_cost

# constraints below;:::

# maxium storage availability
con10 = m.addConstr(stor_m1 <=100, name ='storage_1st_month')
con11 = m.addConstr(stor_m2 <=100,name ="storage_2nd_month")
con12 = m.addConstr(stor_m3 <=100,name ="storage_3rd_month")
con13 = m.addConstr(stor_m4 <=100,name ="storage_4th_month")
con14 = m.addConstr(stor_m1+stor_m2+stor_m3+stor_m4 <= 100, name ='total_storage_capacity_100')
# end of 4th month storage must be zero.
con15 = m.addConstr(stor_m4 == 0, name ="month_4th_zero")
```

```
# regular production in each month must be last 10% of total production of first three month okay
con20 = m.addConstr(r1+r2+r3 >= ((r1+ep1)+(r2+ep2)+(r3+ep3))*0.1)

# CONFUSED WITH IT BETWEEN ABOVE VS BELOW
#    con21 = m.addConstr(r1 >= ((r1+ep1)+(r2+ep2)+(r3+ep3))*0.1)
#    con22 = m.addConstr(r2 >= ((r1+ep1)+(r2+ep2)+(r3+ep3))*0.1)
#    con23 = m.addConstr(r3 >= ((r1+ep1)+(r2+ep2)+(r3+ep3))*0.1)
#    con24 = m.addConstr(r4 >= ((r1+ep1)+(r2+ep2)+(r3+ep3))*0.1)


# regular production less than max production pi okay
con31 = m.addConstr(r1 <= 140)
con32 = m.addConstr(r2 <= 150)
con33 = m.addConstr(r3 <= 140)
con34 = m.addConstr(r4 <= 160)

# Extra production less than max extra production qi okay
con40 = m.addConstr(ep1 <= 50)
con41 = m.addConstr(ep2 <= 75)
con42 = m.addConstr(ep3 <= 70)
con43 = m.addConstr(ep4 <= 80)

# non-negativeity constraints okay
con70 = m.addConstr(r1>=0)
con71 = m.addConstr(r2>=0)
con72 = m.addConstr(r3>=0)
con73 = m.addConstr(r4>=0)
con74 = m.addConstr(ep1>=0)
con75 = m.addConstr(ep2>=0)
con76 = m.addConstr(ep3>=0)
con77 = m.addConstr(ep4>=0)
con78 = m.addConstr(stor_m1 >= 0)
con781 = m.addConstr(stor_m2 >= 0)
con782 = m.addConstr(stor_m3 >= 0)
con783 = m.addConstr(stor_m4 >= 0)

#solve
m.optimize()
```

This is How I wrote an code. I tried to make it clean and I think I followed all of contents that we talked in the meeting. and this is the result from above.

```
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 6 physical cores, 12 logical processors, using up to 12 threads
Optimize a model with 27 rows, 8 columns and 78 nonzeros
Model fingerprint: 0x634cffff
Coefficient statistics:
  Matrix range     [1e-01, 4e+00]
  Objective range  [0e+00, 0e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [5e+01, 2e+03]
Presolve removed 23 rows and 4 columns
Presolve time: 0.00s
Presolved: 4 rows, 4 columns, 13 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    0.0000000e+00   2.886175e+01   0.000000e+00      0s

Solved in 2 iterations and 0.01 seconds
Infeasible model
```

However, How can I make printing optimal values of the objective functions

```
# Print optimal value of the objective function
print('\nTotal Cost: %g' % m.objVal)
# Print optimal values for the decision variables
print('\nDecision variables:')
for v in m.getVars():
    print('%s = %g' % (v.varName, v.x))
```

I copied and paste from RBC ipynb.

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-33-5b5da34f7657> in <module>
      1 # Print optimal value of the objective function
----> 2 print('\nTotal Cost: %g' % m.objVal)
      3 # Print optimal values for the decision variables
      4 print('\nDecision variables:')
      5 for v in m.getVars():

src\gurobipy\model.pxi in gurobipy.Model.__getattr__()

src\gurobipy\model.pxi in gurobipy.Model.getAttr()

src\gurobipy\attrutil.pxi in gurobipy.__getattr()

AttributeError: Unable to retrieve attribute 'objVal'
```