In [1]:

```
from gurobipy import *
```

In [2]:

```
# creeating a model

m = Model("Linear programming")
```

```
Using license file C:\Users\Aloukik Aditya\gurobi.lic
Academic license - for non-commercial use only
```

In [3]:

```

#---- I am taking value of x y z from 0 to 1 which denotes the percent of water taken from
x = m.addVar(lb = 0.0, ub = 1, vtype = GRB.CONTINUOUS, name="Type A water")#-- x = percent
y = m.addVar(lb = 0.0, ub = 1, vtype = GRB.CONTINUOUS, name="Type B Water")#-- y = percent
z = m.addVar(lb = 0.0, ub = 1, vtype = GRB.CONTINUOUS, name="Type c Water")#-- z = percent
```

In [4]:

```
#-- orignally value given was in per liter, but i converted them in price per bottle

m.setObjective(20*x + 30*y + 15*z, GRB.MAXIMIZE)#----------calculated cost for per bottle,
```

In [5]:

```
m.addConstr(2*x + 1.5*y + 0.5*z == 2.5)#-----------adding constraints here

```

Out[5]:

```
<gurobi.Constr *Awaiting Model Update*>
```

Type *Markdown* and LaTeX: α^2

In [6]:

```
m.optimize()
```

```
Gurobi Optimizer version 9.0.2 build v9.0.2rc0 (win64)
Optimize a model with 1 rows, 3 columns and 3 nonzeros
Model fingerprint: 0xdcde53e1
Coefficient statistics:
  Matrix range     [5e-01, 2e+00]
  Objective range  [2e+01, 3e+01]
  Bounds range     [1e+00, 1e+00]
  RHS range        [3e+00, 3e+00]
Presolve removed 1 rows and 3 columns
Presolve time: 0.01s
Presolve: All rows and columns removed
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    5.0000000e+01   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.01 seconds
Optimal objective  5.000000000e+01
```

In [7]:

```
m.getVars()
```

Out[7]:

```
[<gurobi.Var Type A water (value 0.25)>,
 <gurobi.Var Type B Water (value 1.0)>,
 <gurobi.Var Type c Water (value 1.0)>]
```

In [8]:

```
#-- THe result shows that
#--25 percent of water taken from A container
#--100 percent of water taken from B conatiner
#--100 percent of water taken from C container
```

In [9]:

```
print('Obj: %g' % m.objVal)#------------------this is the final and total optimized value
```

```
Obj: 50
```

In [ ]:

```

```