# Exercise 1: Fire Department

Write three classes and practice making calls on them. The classes should build on each other in some cases and use each other in other cases.

### Motor

The motor class has a constructor and destructor that print messages letting you know something is happening. Motor also has:

* start() Print messages if already running or fuel is 0
* stop() Print messages if not running
* getFuelLevel():int Return fuel level (0 through 4)
* isRunning():bool Return true if running
* fillTank() Set fuel level to 4

### Truck

The truck has a motor, but not by inheritance. The truck “uses” a motor. The truck is not a specialized type of motor (inheritance). The truck has a constructor and destructor that print messages. Also:

* start() Pass this request directly to motor
* stop() Pass this request directly to motor
* topOffTank():int Check the motor level and fill if not full and return the amount of fuel used
* moveTo(x:int, y:int) Start motor, move to XY, and stop motor (check conditions as needed)
* getX():int Return truck’s X location
* getY():int Return truck’s Y location

### Fireman

The fireman has a name. The fireman can be assigned a truck (pointer) to drive. The fireman has a constructor and destructor that print messages. Also:

* setTruck(myTruck:Truck\*) Set’s the fireman’s pointer to his or her truck
* removeTruck() Set’s the fireman’s pointer to null (nothing)
* handleFire(x:int, y:int) Move truck to X,Y and print message
* returnToBase() Move truck to 0,0 and refuel

Write test functions for each object in turn. You can write straight code to access the objects or create a user input loop and ask the user what to do (like 1=start, 2=stop, etc).

Have fun with this exercise. Add your own methods and features to the objects. Use lots of prints so you can see what is happening.

# Solution:

This is one big file. A better solution would have these classes broken out to separate files.



















