# Task Objective

Your task today is to develop your knowledge of advanced class structures and understand how to use them to ensure encapsulation of your code.



Exercises  
Save different versions for each exercise below so you have a copy of all your work.

1. One to Many

Implement two separate classes for the example above and write a test program to instantiate the classes.

1. Member Class

Now the only purpose of the manager class is to hold information about the programmers they manage. Implement the Programmer class as a member object of Manager. Write a method in the Manager class called printSalaries() this prints a report of all the salaries.

1. Local Inner Class

Before the salaries are printed permissions need to be granted. Create a local inner class within printSalaries() called Permission that has one method – grantPermssions(). This can simply print “Granting Permsssions to print Salaries….”.

Ensure you instantiate the inner class and call the method grantPermissions().

1. Anonymous Class

Alter the code above so that this is an anonymous class – remember to save both versions of your code.

1. Enum Class

This is a special type of class that can be used to contain values that serve as constants.

So by using Enum Classes you can guarantee the correctness of your choices, you'll essentially have localized the spelling of the options in a single class and then use those options everywhere else in the application.

Research how to implement an Enum class and use this knowledge to define ProgrammerTypes – in this example you can have three programmer types. Include these programmer types in your code above.

* SeniorProgrammer
* JuniorProgammer
* TraineeProgrammer