# Defining Your Own Classes

## Overview

In this lab, you will write an application that defines and uses an Employee class. You will define methods and instance variables for the class, and create some instances of the class in the client code.

## Source folders

Student project: StudentDefiningClasses

Solution project: SolutionDefiningClasses

## Roadmap

There are 4 exercises in this lab, of which the last exercise is "if time permits". Here is a brief summary of the tasks you will perform in each exercise; more detailed instructions follow later:

1. Defining a class
2. Creating objects
3. Overloading methods
4. Going further with object orientation

## Exercise 1: Defining a class

Write an application that defines a simple Employee class and creates some instances.

Suggestions and requirements:

* The Employee class needs to hold the name and salary of the employee, and the date he/she joined the company.
* The class must honour the OO principle of encapsulation, so make sure the instance variables are private. Define public getter and setter methods if you need them.
* The class needs to allow an employee to have a pay raise, so define a payRaise() method that takes the amount of the pay raise and adds it to the employee's current salary.
* The class should also have a toString() method that returns a textual representation of the employee's info.

## Exercise 2: Creating objects

Define a separate class, where you can create some Employee objects and invoke methods upon them. Exercise the methods fully, ensuring that they all work correctly.

**Exercise 3: Overloading methods**

In the Employee class, define a few overloaded versions of a payBonus() method.

* One version of the method can take no parameters and add a fixed percentage of the employee's salary (e.g. a 1% bonus).
* Another version of the method can take a double parameter that specifies the percentage of the bonus.
* Yet another version of the method can take three double parameters that specify the percentage of the bonus, along with a minimum and maximum salary (such that the bonus only applies if the employee's salary is within that range).

**Exercise 4 (If time permits): Going further with object orientation**

Define a suite of additional classes to represent related information about an employee. For example an employee has:

* A home address and a work address
* A home telephone number, a work telephone number, and a mobile telephone number
* A spouse (optionally)
* Personal information such as date of birth, place of birth, tax code, etc.