

# Semester 2, Week 12 – eTest– 6<sup>th</sup> April 2018

---

You must do this exercise completely on your own on a College computer. You must not communicate with any other person (other than the invigilators) in any way. You have 50 minutes in which to address this problem and submit the solution on Blackboard. You can only use:

- Eclipse or any other development environment on the College machines.
- A web browser. You may only access <http://mymodule.tcd.ie/> (the Trinity Blackboard system) to view the coding standard and/or to submit your program at the very end of the session.

## The Problem:

Download the following from [goo.gl/w6JiE2](http://goo.gl/w6JiE2) and use it in your program:

- 1) Employee.java; 2) EmployeeAppraisal.java; 3) HRClient.java; 4) HRCodes.java

Using the above classes and Interface, develop a program for a Human Resources department that supports 1) the addition of employees; 2) viewing all employees; 3) finding an employee, 4) recording an employee appraisal, and 5) promoting an employee. To develop this program, create a Java class called **HRDepartment**, which contains a **collection, called employeeList** (stores objects of type Employee) and the following instance methods (non-static):

**1. public void addEmployee(int employeeId, String name, int role)**

This method is required to create an Employee object using the *parameters* provided and add it to the employeeList collection.

**2. public void printAllEmployees()**

This method is required to loop through the employeeList collection and print each employee details (employeeId, name, promotionDate, role & appraisalHistory).

**3. public Employee findEmployee(int employeeId)**

This method is required to loop through the employeeList collection and return the Employee object which has the same *employeeId* as the one provided or *null* if no Employee is found.

**4. public boolean recordEmployeeAppraisal(int employeeId, Date appraisalDate, int score)**

This method is required to a) find the Employee with the given *employeeId*, b) create an EmployeeAppraisal object using the *appraisalDate* and *score* provided, and c) add the EmployeeAppraisal object to the appraisalHistory collection belonging to the Employee. True is returned if successfully recorded and false otherwise (e.g. if the Employee is not found)

**5. public boolean promoteEmployee(int employeeId)**

This method should promote a non-executive employee if he/she has exceeded expectations in both of the last two appraisals and has not been promoted in the past year. To do this you are required to:

- find the Employee with the provided *employeeId*,
- check the last two appraisal scores in the Employee appraisal history collection ensuring both are “exceeded expectations” (use the HRCodes interface for constants),
- ensure that the last promotion was at least a year ago,
- if (b) and (c) are satisfied, update the employee role as follows (a tester is promoted to a developer, developer to a designer, designer to a manager, and manager to an executive [use the HRCodes interface for constants]), set the promotion date to today’s date, and return true. *PTO for sample code to get the current date and last year’s date.*

**Submit HRDepartment.java on Blackboard using the eTest 4 link**

Getting the date from 1 year ago	Getting today's date
<pre>Calendar calendar = Calendar.getInstance(); calendar.setTime(new Date()); calendar.add(Calendar.YEAR, -1); Date date = calendar.getTime()</pre>	<pre>Calendar calendar = Calendar.getInstance(); calendar.setTime(new Date()); Date date = calendar.getTime()</pre>