

# The University of Nottingham

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

A LEVEL 2 MODULE, AUTUMN SEMESTER 2013-2014

## SOFTWARE ENGINEERING DESIGN

Time allowed 2 HOURS

---

*Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced*

***Complete ALL the examination tasks***

**You may use either the CODEBLOCKS or MICROSOFT VC compilers**

**Please note well: for all tasks marks will be given for the correct declaration of classes and functions even if their definition (implementation) is missing.**

**You may choose to provide either separate files for each of the tasks or a single solution for the whole exam - just be clear please.**

*Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.*

***DO NOT turn examination paper over until instructed to do so***

**ADDITIONAL MATERIAL:      NONE**

### **Note to Invigilators:**

This is an open book examination and candidates may bring in any material on disks, memory sticks etc that they wish.

The course material must be available to the candidates on the computers.

Internet and email access must be disabled on the computers and not available to the candidates.

**Introduction:**

A *railway company* operates a number of *freight trains* on a rail network. Each *freight train* carries a number of cargo containers. The company requires the development of a C++ software package that will allow it to manage its *freight trains* in order to meet the demands of its customers in the most cost effective manner.

Each *freight train* is to be characterised by (i) the *maximum number of cargo containers* that it can carry, (ii) the actual *number of cargo containers* it has been loaded with so far.

**Examination Task: part 1 (40%)**

Your first task is to develop the basic *freight train* class. Please provide just the minimum functionality expected of any well designed class. Please make any required design decisions in the context of the project described above and in accordance with good software engineering practice.

**Examination Task: part 2 (10%)**

Provide stream operators for the *freight train* class.

**Examination Task: part 3 (15%)**

Add to the *freight train* class a member function called, *try\_to\_load\_cargo\_containers*, which accepts an integer argument stating the number of additional cargo containers to try to load onto the *freight\_train*. The *freight train* is to be loaded with as many of these cargo containers as it can accept and the function is to return an integer giving the number of cargo containers that could not be loaded because the *freight train* became full.

**Examination Task: part 4 (15%)**

Write a main program that:

- 1) Creates an array of *freight\_trains* dynamically, reading their data from a file of your creation. The format of this file is your engineering decision as is a set of suitable test data.
- 2) Asks the user to enter via the keyboard how many cargo containers are to be loaded in total onto these *freight\_trains*.
- 3) Distributes this total number of cargo containers amongst the *freight\_trains*. Here, it can be assumed that cargo containers are allocated to the *freight\_trains* in the order that the *freight\_trains* were read from the file.
- 4) Reports to the screen how the cargo containers have been distributed amongst the *freight\_trains*

### Examination Task: part 5 (15%)

To provide for future expansion, it is now considered appropriate to generalise to the concept of a *train* which has *freight\_train* as one of many possible sub-categories.

All *trains*, of whatever sub-category, are to contain the member variable *maximum\_speed* in mph, but only *freight\_trains* contain the member variables *maximum number of cargo containers* and *number of cargo containers*.

Please modify your code to reflect the relationship between *train* and *freight\_train*.

In particular please provide suitable constructor, copy constructor, destructor and assignment operators for both *train* and *freight\_train*.

### Examination Task: part 6 (5%)

All *trains* of whatever sub-category, are to contain a member function called *get\_maximum\_speed*.

The default behaviour of *get\_maximum\_speed* is return the value of the variable *maximum\_speed*.

However for *freight\_trains* in particular, it is recognised that the maximum speed of the train will probably depend upon how much cargo has been loaded.

At this stage in the design we do not know how this calculation will be implemented for *freight\_trains*, but please ensure that the design would permit a different implementation of *get\_maximum\_speed* to be provided for each different sub-category of *train* in the future.