This paper contains questions designed to test both your understanding of core programming concepts and your ability to demonstrate that understanding whilst writing short programs (or parts of programs). For the sake of clarity, you should note that the Java language should be used whenever a question asks you to either write code or provide examples. On that basis, wherever the paper asks you to provide one or more programs, classes, methods, statements, expressions, fields, etc. your answers should conform to the syntax rules of the Java language as taught in this year's CM10227 lectures (i.e. Java version 7).

1. (a) A Java program is required to record information describing the coders, customer supporters and office managers working at a software company. Provide such a program to satisfy the requirements, below.

- As you do so, note that:
  - you need not provide a more complete implementation of the program than that which is explicitly specified below.
  - for example, you do not need to write body code when providing methods
  - you should use at least one fully implemented *class*, at least one *interface* and at least one *abstract class* in your answer. You may use more than one in each case.
- For each of the following roles, create a *class*, *interface* or *abstract* class as appropriate:
  - employee, coder, team leader, software engineer, trainee, customer supporter, junior technician, senior technician, office manager.
- Whilst creating these *classes*, *interfaces* and *abstract classes*, also note that:
  - everyone is an employee.
  - team leaders, software engineers and trainees are coders.
  - junior technicians and senior technicians are customer supporters.
  - team leaders and office managers supervise.
  - team leaders, software engineers and trainees write code
  - junior technicians and senior technicians both log bugs but the
  - process for logging bugs differs for each role
- You should also add data *fields* at appropriate points to capture the following information:
  - every employee has a name and an employee number.
  - every coder has a specialist language.
  - office managers have a spending limit
- Finally, add methods for getting and setting the content of these fields.

[10]

[1] for each of the following (max 5)
– Stub of Abstract Medic class (or alternative Abstract Class)
– Correct extension of that abstract class to other subclasses
– Stub of Supervising Interface (or inherited 'sells method")
– Use of inheritance for

doctors and nurses (or sutiable alternative inheritance)
– Provision of at least one concrete class

[0.5] for each of the following class definitions with constructors:
– consultant, registrar, matron,
staff nurse, ambulance crew member. (max 2.5)

[0.5] for setters and getters to handle
– name/address/national insurance number,
specialism, rank, qualification, department, ambulance (max 2.5)

(b) x [4]

x

(c) x [2]

x

(d) x [2]

x

(e) x [2]

x

(e) x [2]

x