

THE BRITISH COMPUTER SOCIETY

THE BCS PROFESSIONAL EXAMINATIONS
Diploma

OBJECT ORIENTED PROGRAMMING

28th April 2006, 2.30 p.m.-4.30 p.m.

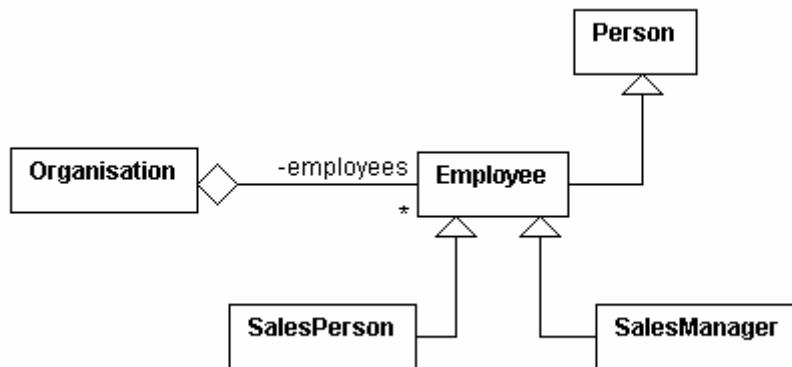
Answer FOUR questions out of SIX. All questions carry equal marks.

Time: TWO hours.

*The marks given in brackets are **indicative** of the weight given to each part of the question.*

Important Note for Questions 1, 2 and 6.

If you choose to answer questions 1, 2 or 6 then you should make use of the following Unified Modelling Language (UML) class diagram. It models a company that employs sales staff and sales managers.



Unified Modelling Language (UML) class diagram.

1.
 - a) Using the UML class diagram given at the start of the examination paper, construct a UML object diagram showing one SalesManager and two SalesPersons working for the same Organisation. Give a short explanation of the diagram. **(5 marks)**
 - b) If the two SalesPersons and the one SalesManager are employed by the same Organisation, then explain whether there is a need to distinguish between these differing types of employee. **(5 marks)**
 - c) Revise the above class diagram to introduce a Secretary class, representing an employee not involved in any sales activities. Explain the principal revisions that have been made to the diagram. **(5 marks)**
 - d) Revise the above class diagram so that a SalesManager is given managerial responsibilities for a team of sales staff. In your scheme explain how a management hierarchy of SalesManagers would be possible. **(5 marks)**
 - e) Revise the object diagram from part 1a) showing both SalesPersons being managed by the SalesManager. **(5 marks)**

Turn over]

2. a) What do you understand by the term *class hierarchy*? (4 marks)
- b) What do you understand by the term *abstract class*? (4 marks)
- c) Draw a revision to the UML class diagram given at the start of the examination paper, clearly distinguishing those classes that have been changed into abstract classes. Explain why they have changed. (6 marks)
- d) Offer a strong argument why object oriented software should be developed in terms of abstract classes. (6 marks)
- e) Using a programming language of your choice and making any reasonable assumptions, give an outline of the implementation of an abstract class from part 2 c). (5 marks)
3. a) A guiding principle for object oriented development processes is that they should be:
 i) Use-case driven
 ii) Iterative and incremental.
 Explain what is meant by these terms. (6 marks)
- b) How does architecture-centric development make the system's architecture the primary focus? (3 marks)
- c) How does an iterative and incremental development help minimise risk when developing a system? (3 marks)
- d) Discuss how an iterative and incremental development process can be integrated with the testing activity. (5 marks)
- e) In the context of object oriented development, explain what is meant by the terms:
 i) Functional testing
 ii) Unit testing (8 marks)
4. a) Give definitions of the following:
 i) abstract data type
 ii) encapsulation
 iii) structured programming
 iv) coupling
 v) cohesion (15 marks)
- b) Choose THREE of the above concepts and discuss how each has contributed to the development of object oriented programming. (10 marks)
5. a) Briefly explain what is meant by the term, *Design Pattern*. (3 marks)
- b) Discuss two major advantages of using *Design Patterns*. (6 marks)
- c) Discuss two major disadvantages of using *Design Patterns*. (6 marks)
- d) Describe a *Design Pattern* with which you are familiar. Your answer should include the motivation for the existence of the *Design Pattern*, its structure, participants and consequences of its use. (10 marks)

6. a) A requirement of object oriented systems is to manage a collection (or container) of objects, e.g. an array or a set.
- i) Identify a collection of objects in the UML class diagram given at the start of the examination paper.
 - ii) Give one example of a collection class (other than the **Vector** class described below) with which you are familiar.
 - iii) Most collection classes are described as *generic*. Explain what is meant by this term. **(12 marks)**

Consider the following scenario:

*A class **Vector** is a collection (container) class that grows dynamically as elements are added to it. Each element has a unique index associated with it. Indices start at 0 and increase by 1. For example, the first element has an index of 0, the second 1 and the third 2.*

*To add an element to a **Vector** we supply an index and the element to be added. For example, `add(2, element)` adds element to the third position in the **Vector**.*

*To return an element from a **Vector** we only need its index. For example, `get(0)` returns the element at the first position in the **Vector**.*

*Similarly, to delete an element we supply the index. For example, `removeAt(1)` deletes the second element from the **Vector**.*

- b) You are required to use the **Vector** class in the construction of a **Stack** class. This new class should mimic a stack. Elements can be added to the top of the stack with a **push** method and retrieved from the stack with a **pop** method. The latter also deletes the element from the stack. Crucially, it should not be possible to add or retrieve elements at intermediate positions in the stack.

Discuss in detail the problems associated with using specialisation (inheritance) to develop the **Stack** class from the **Vector** class. Again, you should use a programming language of your choice to illustrate your answer. **(7 marks)**

- c) Suggest a better alternative to the use of specialisation and explain how it might be implemented. **(6 marks)**