THE BCS PROFESSIONAL EXAMINATION
Diploma

April 2000

EXAMINERS’ REPORT

Systems Design

Question 1

a) ***Describe the object-oriented approach to systems design.*** ***(18 marks)***
b) ***State the main differences between this and other traditional approaches.*** ***(7 marks)***

Answer Pointers

The candidate should describe an object-oriented approach with which they are familiar. This might be UML. Activities such as use-case diagram development, class diagram production and dynamic modelling using state diagrams and sequence diagrams can be discussed. Each of these areas involve a number of concepts and techniques so outline description only is expected.
(18 marks)

As far as the difference between object-oriented and other traditional approaches is concerned candidates can discuss:

the iterative (onion-like) approach as compared to the waterfall approach

- the merging of analysis and design as compared to distinct stages
- the perceived improved representation of real-world objects as opposed to computer records
- the use of the same concepts throughout the refinement process
- the reusability of objects
- the ability to address a wider range of application domains.

(7 marks)

Examiners' Guidance Notes

For the first part of this question, most candidates concentrated on the class diagram and described the components therein. This was generally done well and in view of this, although the examiner originally expected other techniques also to be discussed, for most candidates most of the marks to be given for this part of the question were assigned to the coverage of the class diagram. Additional credit was given however if candidates mentioned other techniques in object-oriented systems development. In future candidates should be aware that class diagram production, although important, is not the only technique in object-oriented design. A number of candidates ignored the second part of the question and therefore lost marks. Candidates should be advised to address all parts of a question.

Question 2

Consider the following scenario.

A building firm carries out various jobs for customers. A customer may be a company or a private individual. The building firm records the name and address for each customer. For companies, it also records a list of contact-names together with their department and telephone numbers. For private individuals, the firm records a daytime and evening telephone number.

The firm records a description, start-date and estimated duration for each job. A job consists of various items of work. Each item of work has a description, a labour cost in man-hours and a list of materials together with quantities required and prices.

The firm needs a database system to record this information. The database system should be able to carry out various functions including the following.

Produce a bill of materials for a particular job.
Calculate the overall cost of a particular job.
Add and delete contact names for a company customer.
List items of work and jobs that use a particular material.

Using a method with which you are familiar, create a class diagram that represents the above scenario. ***(25 marks)***

Answer Pointers

Likely classes are *customer*, *company*, *private individual* and *job*.

The *job* class may be broken down further into *item of work* and *material item* .

Suitable relationships should be shown between the classes and suitable attributes and operations should be given. There is a generalisation relationship between *customer* and its subclasses *company* and *private individual.* This should be shown with suitable definition of attributes and operations.

An appropriate notation such as the UML class diagram should be used.

Examiners' Guidance Notes

This question was generally answered well. The examiner's impression was that candidates had been taught and had learnt this material well. The development of class diagrams should continue to be practised as it is at the core of the object-oriented approach. Well done everyone!

Question 3

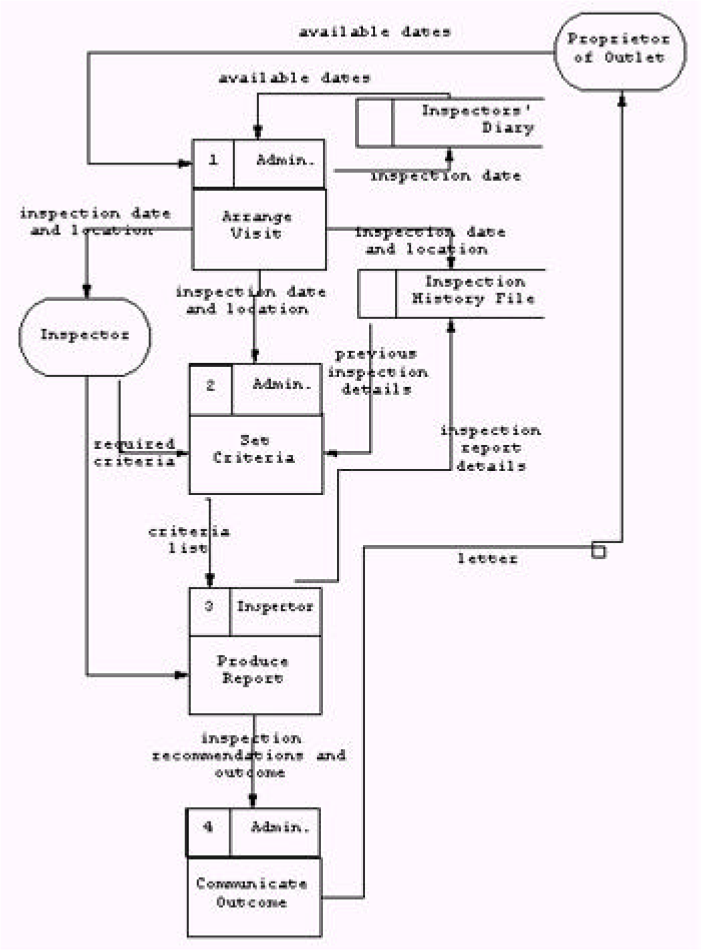
a) *Describe what is meant by the following two terms.*
transform analysis
transaction analysis *(7 marks)*

b) *Consider the following outline dataflow diagram which represents an inspection process for food outlets. The process runs as follows.*

The local authority administration team arranges a date for the inspection visit by consulting the inspectors' diary and consulting with the proprietor of the food outlet. The inspection date and location is communicated to an inspector. The administration team draw up a list of criteria for the inspection based on any previous vist of the outlet and the advice of the inspector. The inspector carries out the visit and produces a report which contains recommendations and an outcome. The recommendations and outcome are then communicated through the administration team to the proprietor of the outlet.

Convert the dataflow diagram into a structure chart using a suitable process.

(18 marks)



Answer Pointers

a) Transform analysis is used to convert a transform-centred data flow diagram into a structure chart. The processes involved in the central transform should be identified from the data flow diagram. Other processes will usually be involved in input or output and will be represented as input or output modules

Transaction analysis is used to convert a transaction-centred data flow diagram into a structure chart. The transaction centre must be identified form the data flow diagram and is shown as a high level module with a diamond shape to indicate choice. Other processes will be modules in the structure chart to which control will be directed depending on the choice input to the transaction-centre module.

(7 marks)

b) A suitable structure chart should be drawn. The following principles should be followed.

Create an overall control module
Each process becomes a module in the structure chart
Each flow becomes a coupling
Add control flags where needed to pass condition information
Group main input and output modules together under a higher level module.
Decompose modules that perform more than one function

For instance:

An overall control module should be created e.g. *inspection-system*. Under this will be four modules say *arrange-inspection-visit*, *set-inspection-criteria*, *produce-report*, *communicate-outcome*. The modules *arrange-inspection-visit* and *set-inspection-criteria* may be grouped together under a module such as *make-preliminary-arrangements*. These processes may break down further. For example *arrange-inspection-visit* may break down into *read-inspectors-diary*, *get-proprietor-availability*, *set-date* and *output-date*. Flows should be shown as suitable couplings. Complex flows may be broken down into simple couplings.

(18 marks)

Examiner's Guidance Notes

Not many candidates knew the answer to the first part of the question although the difference between the two analyses is made clear in the set text. Regarding the second part of the question, candidates, generally produced a suitable structure but often the data flows shown as couplings were omitted. Some candidates however were unfamiliar with the concept of a structure chart and produced diagrams bearing more resemblance to flow charts.

Question 4

An automobile dealership is preparing to transfer its customer sales ordering system from one that required transcription from a clerical form to one with direct input from an on-screen form.

The main details required are Order number, type of car, model, customer details, and up to six optional extras per car. Costs of the car and the options are totalled to give total cost to the customer. Insurance can be provided at special rates to customers. Insurance costs, however, always relate to the age of the driver.

Each car is supplied with a two-year parts and labour warranty

The present clerical form follows:

ABC Motors
New Vehicle Order Form

Order No					
Customer Details **Name** **Address** **Tel No.**					
Model					
Engine Size (tick)	1.6 litres	1.8 litres	2.0 litres	2.5 litres	
Basic Cost					
Colour (Select from dealer list)					
Options (Select from dealer list)					
Option Name				**Cost**	
Registration Number.					
Insurance Option (first named driver is assumed to be the vehicle owner unless otherwise stated) (ALL drivers are assumed to be resident at the owner's address)					
Driver	**Date of Birth**	**Convictions, etc.**	**Loading**		
			Insurance Total		
				Cost Total	
				VAT	
				Customer Total	

(a) Identify the reasons for undertaking a normalisation process. ***(5 marks)***

(b) Normalise to the third normal form, showing all stages, the attributes (shown in bold face) for this un-normalised order form. Provide a full explanation of your decisions, details of any

checks carried out during the process and identifying any key fields.
(15 marks)

(c) Suggest the major problems likely to be encountered during the implementation of such a form when using a modern Fourth Generation Language (4GL), taking into account that the screen form layout must simulate the clerical document as closely as possible.
(5 marks)

Answer Pointers

(a)

- Avoid duplicated records
- Enables relational Dbs to be implemented
- Enables single table updating
- Identifies relationships

(b)

UNF	1NF	2NF	3NF
Order No. *Customer Details* Name Address Tel.No. Model Engine size Basic Cost Colour *Options* Option descrip’n r Option price r Registration No. *Insurance* Name of driver r Insurance Group r Convictions r Loading r Insurance cost r Total Cost VAT Customer Total	**Order No**. *Customer Details* Name Address Tel.No. Model No. Engine size Basic Cost Colour Registration No. Total Cost VAT Customer Total **Order No**. Option descrip’n Option price **Registration No** Name of driver Insurance Group Convictions Loading Insurance cost	**Customer No.** Name Address Tel.No. **Order No**. CustomerNo. Model No. Engine size Basic Cost Colour Registration No. Total Cost VAT Customer Total **Order No**. Option No. **Option No.** Option descrip’n Option price **Registration No.** Name of driver Insurance Group Convictions Loading Insurance cost	**Customer No.** Name Address Tel.No. **Order No**. CustomerNo. Model No. Engine size Basic Cost Colour Registration No. Total Cost VAT Customer Total **Order No**. Option No. **Option No.** Option descrip’n Option price **Registration No.** Driver No **Driver No.** Name of driver Insurance Group Convictions Loading Insurance cost

Note: Use of artificial keys which would not necessarily be seen by the end-user. Third normal form removes calculated fields. Registration No. is specific to car and drivers. Customer name & address is a separate table as customer may purchase other vehicles.

(c) Forms and subforms required to input data to more than one table
Some 4Gls make replication of clerical documents difficult

Examiners Guidance Notes
The three stages of normalisation must be identified and shown, i.e isolation into separate tables of repeating groups (1NF), isolation of those fields not totally dependent on key fields (particularly composite ones) (2NF), and removal of those fields related to other fields within the table records (such as calculated fields) (3NF).

Each car may have more than one option, and may have more than one driver. The customer has only one address, but may buy more than one car from that company.
The option description and price are dependent on the option and not the order number.
The total cost and customer total are calculated fields and can be calculated within queries as required

The principles employed are that data wherever possible, should appear only once, so that for example, if a customer's address changes, then it need be changed once only, and not in every occurrence of the customer's record.(the customer's id. number does not change, and provides the link)

Candidates who justified their choice of tables in a realistic way received the appropriate marks, as it is very difficult to provide a definitive normalisation in real life, and even more difficult in a limited case study, as much depends on the use and environment of the data.

Question 5

Application development tools and methods have become more important as software development costs increase.

Write short notes identifying the roles of the following in application generation.

(a) Java	***(5 marks)***
(b) SQL	***(5 marks)***
(c) HTML	***(5 marks)***
(d) C++	***(5 marks)***
(e) XML	***(5 marks)***

Answer Pointers and Examiners Guidance Notes

(a) Java
Portability
Object-oriented
Distributed
Reliable
Secure
Needs interpreting
High performance
Multi-threaded

Examiners Guidance Notes
Experienced candidates pointed out that provided that the Java VM (virtual machine) existed on a particular computer, then the Java script was machine independent

(b) SQL
Simple but powerful English language based
Used for data definition and manipulation in relational database systems
Provides flexibility in larger 4GL systems
Set based, non procedural

Examiners' Guidance Notes
The importance of SQL is in providing a standard, structured, language for data definition and manipulation in relational databases. More experienced candidates appreciated this importance and gained extra marks

(c) HTML
Original language for web pages
Related to SGML
Now has better derivatives for advanced web page design
Unwieldy in basic version
Limited without use of web page generator software

Examiners' Guidance Notes
Many candidates appreciated the importance of HTML to the Internet in general, while better candidates appreciated its limitations in its basic form, which has lead to modern developments. Mention of both web page generators, and their effect on end-user web page development, and of the use of browsers, (like word processor packages), mentioned by some candidates, showed an understanding of the basics of HTML code
Many candidates are to be congratulated for their use of examples to reinforce their answers

(d) C++
Object-oriented derivative of C
Too powerful for inexperienced users
Can have flaws when used in very large systems
Requires complex programming skills
Widely used for writing system software
Used for object-oriented systems
Successful candidates pointed out that C++ was platform dependent.

Examiners Guidance Notes
It was not enough to say what C++ was, a summary of C++'s place in the range of tools available for software development was also needed. Again the use of examples is to be congratulated.

(e) XML
XML is a method for putting structured data in a text file
XML is a set of rules, guidelines, conventions for designing text formats for data, in a way that produces files that are easy to generate and read by computer.
Allows experts (such as programmers) to more easily debug applications
A simple text editor can be used to fix a broken XML file.

Based on well-established SGML.

Examiners' Guidance Notes
XTML, like HTML, stems from SGML. This point was understood by some candidates, who also pointed out that it was not a tool for end users, but very useful for software engineers.

Question 6

(a) Describe the role of Entity Life History diagrams in systems design and discuss how it feeds into the development of a system. Evaluate the contribution Entity Life Histories (ELH) can make to the design and implementation of a system. (13 marks)

(b) Produce an Entity Life History chart for the new car entity (in the automobile dealership in Question 4), from the details given below justifying the details and events included.

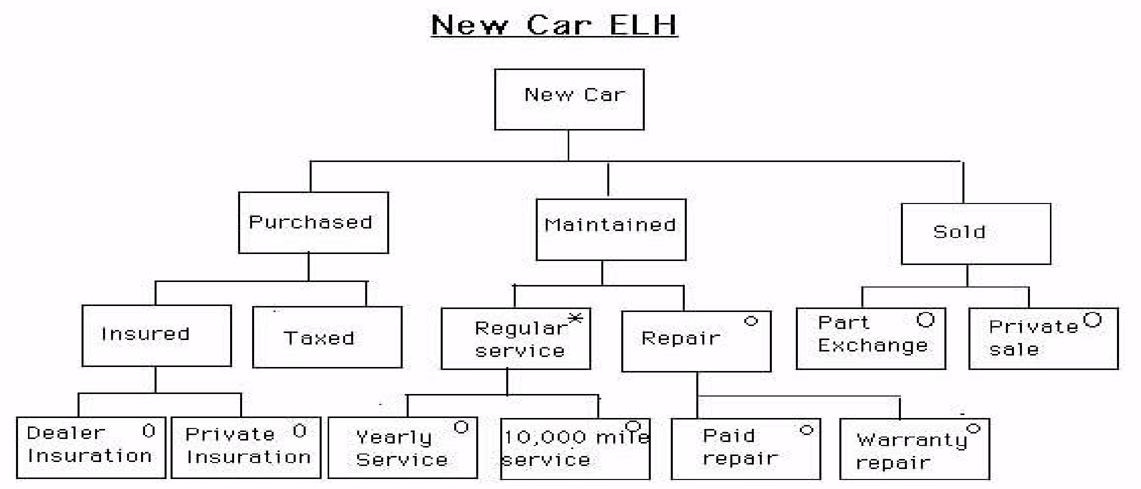
(12 marks)

When a new car is purchased it must first be insured and then taxed, before the customer can complete the transaction. Insurance can be provided by the car dealership or privately by the customer.
The car must be regularly maintained to comply with the conditions of the warranty. Service intervals are 10,000 miles or 12 months, whichever comes first. During the first two years of ownership, servicing is free.
Most of the components of the car are repaired free under warranty, but there are a few which the customer must pay for.

When the customer sells the car, he may decide to part-exchange it for a new car, or sell it privately.

Answer Pointers
(a) ELHs
Document all events that can effect the life of an entity
Events cause updates (effects)
Shows sequence of events over life of an entity
Can show iteration
Status codes, sequence of events can feed into Function matrix, ECD
ELH factors
User requirements
Identify events
Up-pass analysis of entity behaviour
Down-pass analysis of entity behaviour
Look for death events
Events causing creation/deletion of an entity
Events setting values of non-key attributes
Events creating/deleting occurrences of relationships
Events causing changes to state of an entity

(b) Assumptions made
Customers can decide not to include insurance from the dealer
Warranty repairs may not be needed
Regular servicing is a pre-requisite of the warranty



Examiners Guidance Notes

Entity Life Histories (ELHs) fill a niche in the range of documentation tools available, particularly for SSADM, when the sequence of use and life of an entity is needed to complement the other information available. Like all documentation tools they are not obligatory, but used when needed.

Better candidates identified the ways in which Entity Life Histories complement the other documentation tools by providing useful information that could not be shown by other means.