THE BCS PROFESSIONAL EXAMINATION
Diploma

April 2001

EXAMINERS’ REPORT

Systems Design

The standard of papers this year was disappointing compared to previous years. A major fault was the poor examination technique demonstrated. Often answers did not address the question as put and also parts were missed, e.g. question 2 “compare and contrast” often resulted in a description of the two techniques only, thus wasting marks. Many students seemed to be badly prepared as they were able to attempt one or two questions fully but subsequent answers ‘ran out of steam’ and were very sparse and of poor quality. Although the syllabus states a structured method and an object oriented method need to be known, student preparation in OO methods was poor. There are some very good books on UML on the reading list and students should refer to these. *[A new book on UML has been added to the 2001/2002 Systems Design reading lists].*

QUESTION ONE

a) Identify the important features in good screen menu and dialogue design (8 marks)

b) Suggest a screen menu and dialogue structure for the following situation: -

A mail-order processing system is being converted from a batch processing system, where a number of operatives each undertake one clerical process in the chain of operations, to a logical screen based system where each operative handles all clerical operations.

The process is as follows: -

Customers can order goods from the mail order system by e-mail response from an Internet web-page form or by telephone. Credit worthiness is checked against company records for account customers or with the credit card company for credit card orders. A stock check is carried out and the customer informed if any item is not in stock and given the options of waiting, selecting an alternative, or cancelling the order for the out of stock item. When the order is agreed the details are passed to the warehouse where items are selected, packaged, delivered to the customer by courier. The delivery note, signed by the customer is acknowledged within the system and the customer is invoiced in the case of current account customers. (17 marks)

Question 1 was in line with the CPD objectives of the BCS. Menu systems are in a constant state of development, and candidates must be aware of trends and developments. The principles of good menu and dialogue design must be understood and applied to avoid a range of errors ranging from simple user frustration to potentially life-threatening situations in safety-critical systems.

Candidates could have made more of part b) by use of practical experiences, both good and bad, of the wide variety of menus, interfaces and dialogues in current use.

Candidates who gained the highest marks applied the principles of part a) to the second part of the question, using practical experience to provide both a menu structure and adequate dialogue. A structure implies a hierarchy of some kind.

Many candidates did not make it clear in their answers whether the users of the customers on the Internet or operatives within the mail order company.

The principles of menu navigation were generally not followed.

Answer Pointers

a) Suggested guidelines for good menu and dialogue design

1. Minimise keystrokes

2. Menus should have meaningful titles

3. Menu items, including icons, should be clear and unambiguous

4. Menu organisation should relate to user tasks

5. Menus/ dialogues should be concise

6. Consistent menu style

7. Dialogue should provide adequate responses to input data

8. Dialogue should have help feature

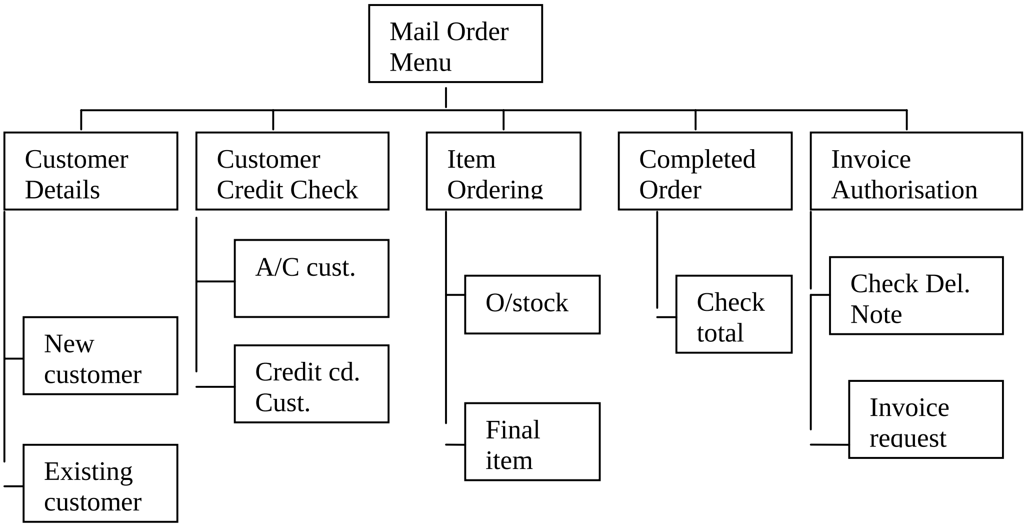
9. Dialogue structure should be consistent across tasks

10. Users should be defined

11. On-line help should be adequate

12. Navigation between menus/screens should be clear and unambiguous.

b) A Menu structure could be as follows:-



Any other valid diagram would be acceptable. Candidates must demonstrate an awareness of the principles in part a).

User dialogues should show both inputs and output responses, e.g input of order item no. should produce one of two responses; ‘Quantity OK’, or ‘Quantity insufficient- contact customer’

Marks Breakdown

a)	Each point	(1 marks up to 8)
b)	Valid structure	(5 marks)
	Suitable menu section details	(6 marks)
	Dialogue - to match menu sections	(6 marks)

QUESTION TWO

a) Compare and contrast the following two process design techniques:-

1. **Decision Tables**
2. **Structured English** **(10 marks)**

b) Produce the Structured English process description for the activities described in Question 1 above. (15 marks)

Part a) of the question set out to examine the candidates' knowledge of two methods of documenting systems, how they differ, and how they complement each other in different situations.

Few candidates understood the situations where one or other of the methods was preferable.

Part b) was used to assess the knowledge of one of the documentation methods contrasted in part a). Few understood the principle of iteration, for example, where items are selected in turn, availability assessed, until the last choice of items has been made.

A great deal of allowance was made for variations in Structured –English methods, and marks allocated where the answer was structured and logical.

Answer Pointers

a) 1. <u>Decision Tables</u>

1. Decision tables enable IF – THEN rules to be covered in diagrammatic form
2. Can have limited, extended entry, or mixed entry formats
3. Can be structured, improving readability

2. <u>Structured English</u>

1. Provides english-like format for description of processes and decisions
2. Removes "noise" words that can cause confusion
3. Provides for easy conversion to pseudo-code
4. Often easier for situations with loops in processes

<u>Decision Tables</u>	**<u>Structured English</u>**
Better for complicated decision processess	Better for nested processes
Easier to show as structured modules	Easier for file handling processes
	Better for showing sequences & iterations

b) Typical description of Mail-order Processing system could be:-

```
Record customer details
    IF  New-customer
        Add details
        Allocate customer-no.
    ELSE  input customer-no.
Do Credit Check
    IF  a/c customer check customer account
    ELSE  check with credit card company
Record order items
    DO UNTIL Last-item
        Enter Item-no.
                IF valid enter Quantity
                        IF  quantity insufficient
                        THEN contact customer
                ELSE  request Next-item
        IF Last-item
    DONE
Check complete order
    Check total-cost
        IF over credit-limit
                Contact customer
        ELSE  order to warehouse
Authorise invoice
    Check returned-del-note against order
    IF   unmatch
        Adjust order
        Authorise Invoice
    ELSE Authorise Invoice
```

Marks Breakdown

a)	Suitable features given	(5 marks)
	Meaningful comparison	(5 marks)
b)	Structure to match case study details	(7 marks)
	Matching sets of statements e.g IF..THEN.. ELSE	(3 marks)
	Use of structured english statements	(5 marks)

The use of terminology is not presecriptive, and any logical description would be acceptable

QUESTION THREE

Write short notes on the following, stating the relevance of each in systems design:-

1. **Web page frames** **(5 marks)**

2. **Java beans** **(5 marks)**

3. **Applets** **(5 marks)**

4. **Cookies** **(5 marks)**

5. **JPEG images** **(5 marks)**

A number of important but varying aspects of web design and use were covered in this section. Those students obtaining the highest marks were those who understood the principles. These are set out in the answer pointers above.

Web-based systems are crucial in modern business systems and students are expected to keep up with both trends and basic tools used in designing these systems.

Answer Pointers

1. **Web page frames** split the browser window into sections so that some sections can remain static while other sections can change according to information loaded via links. Frames are a vital aspect of web-page design and operation but can inhibit printing from the browser print feature.

2. **Java beans** are reusable software components that can be visually implanted into builder tools. They carry the concept of re-using components long used in conventional engineering into software construction. Their main features are:-

Introspection	analysis of operation of bean
Customisation	ability to alter appearance of bean
Properties	allowing bean to be manipulated within a program
Persistence	allowing altered beans to be restored to original state.

3. **Applets** are programs written in Java that can be embedded into an HTML page in the same way as an image. When the page is transferred to a user's PC the code is activated by the browser's Java Virtual Machine. They provide for complex standard layouts to be downloaded, wthout having to compile them into more complex lower-level code.

4. **Cookies** are small amounts of data that are embedded into the user's PC by a site contacted by the browser so that the next contact with the site enables quicker access to that site. One use in systems is to reduce the time needed to contact regular customers.

5. **JPEG** is a file format that is one of the two normally acceptable in web pages, the other one being GIF (Graphics Interchange Format). JPEG allows colour photographic images to be compressed to an acceptable file size, compared to GIF images. It usually has three quality levels, which can be selected according to the application. The format allows full-colour images to be downloaded from web sites in acceptable time durations.

Marks Breakdown

Description of each feature	(4 marks)
Relevance to Systems Design for each feature	(1 marks)

QUESTION FOUR

A police force wishes to set up a database to hold details of personnel for payroll purposes. Staff are divided into civilian and police staff. For each member of staff the database will hold staff-number, name, address, grade, payroll-point within grade, start-of-employment-date and start-of-current-grade-date.

Each point on a pay grade has its own monthly pay figure. Civilian staff have a security-level-code which indicates the type of classified material that they may handle.

A list of previous grade and points within grades, with start and end dates is also kept for each employee. Also a list of deductions such as union fees and healthcare plans is kept for each member of staff.

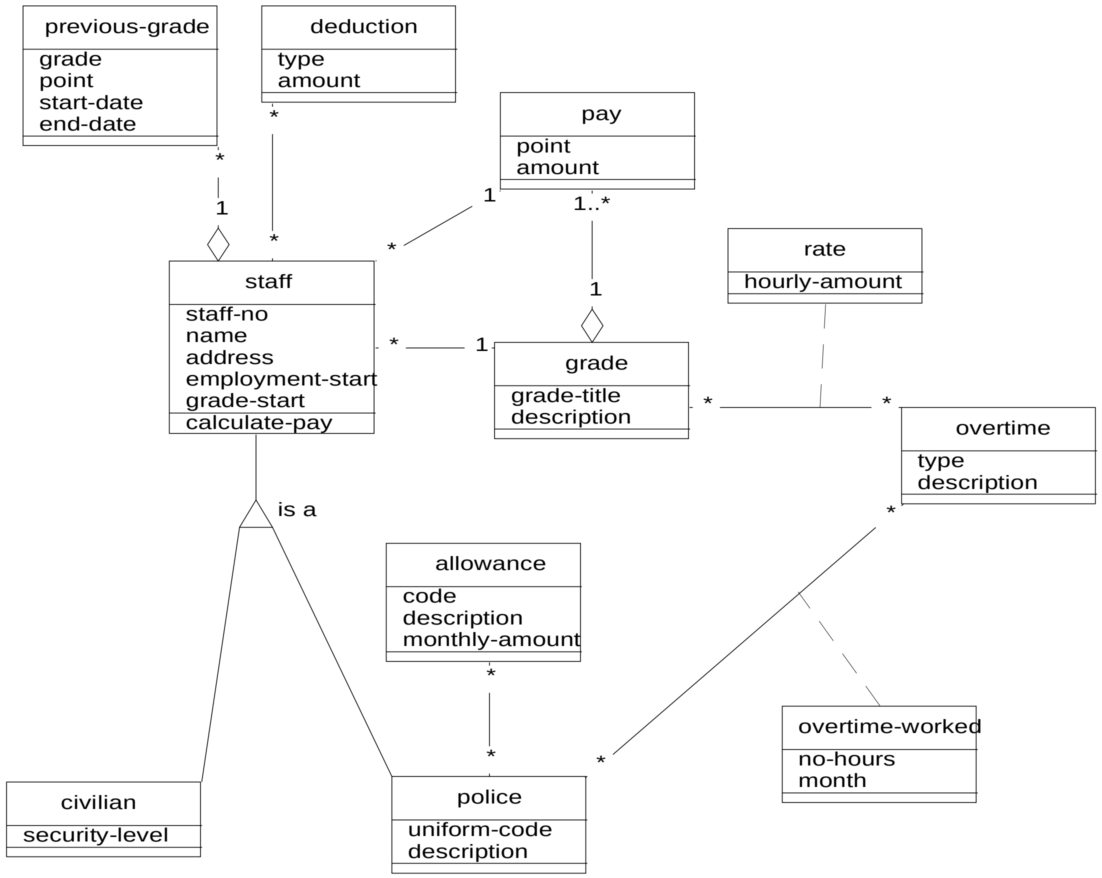
Police staff have a uniform-code, which indicates the type of uniform they should wear. They also have a record of overtime for each month, held as number of hours and type of overtime.

Each police grade has two hourly standard overtime rates: a standard overtime rate; and an extra-overtime rate.

Each member of police staff also has a list of allowances, for instance dog handling, shift or clothing. Each allowance has a code, a description and a monthly amount.

The main operation needed on the above database is to calculate monthly pay. The calculation will vary depending upon the type of employee.

The following is a class diagram for the above scenario.



a) **Explain how a class diagram is, in general, mapped onto a relational model. Consider the concepts of generalisation, association and aggregation in your answer**

(10 marks)

b) **Produce a relational model from the class diagram**

(10 marks)

c) **Provide an outline algorithm for the operation calculate pay in a relational context. State the role of SQL within the algorithm**

(5 marks)

This question generally was not well answered with many candidates scoring under half marks. For part a) most candidates explained the obvious mapping of classes to relations but did not cover the mapping of generalisation and aggregation or the supporting of associations through the use of primary and foreign keys. This was clearly indicated in the question as a requirement and it was the mapping of these concepts for which most marks were available. Some candidates described the identification of such feature as aggregation/generalisation to be mapped, but then failed to state how that mapping was to be made.

Part b) caused some candidates problems in that they failed to understand that a relational model requires a set of normalised tables, which they did not provide. Many produced the tables for each class but did not map the associations, aggregation and generalisation. For example, for the generalisation structure staff-civilian-police many candidates did not post the staff_no into the police-staff relation, which then led to problems in implementing the overtime-worked association.

For part c) many candidates produced an algorithm but lost marks because they didn't state the role of SQL within it as was asked in the question. A common mistake in the construction of the algorithm was the failure to differentiate between the pay calculation for civilian and police employees.

Answer Pointers

a) Principles for obtaining a relational model from a class diagram

For each class:

Create a relation. Determine primary key. If no attribute(s) are suitable for the primary key, create a primary key attribute

For associations:

If one to many

Post the primary key of the uni-part class of the association to the multi-part class in the association.

If many to many

Create a new relation. Make a primary key as a combination of the primary keys of the relations derived from the classes participating in the association

Link classes:

Add attributes of link classes to the new relation created to represent the association. Check the primary key as the new attributes nmay affect it.

For generalisation:

Either post the attributes of the relation derived for the subclasses to that of the superclass. Add a type attribute.

Or

Post the primary key of the relation derived for the superclass to each subclass. A type attribute may be added to the superclass.

For aggregation:

Post the primary key attribute for the relation derived from the subclass to the relation derived for the superclass.

b) A suitable relational schema from the above is :

Staff(staff-no,name,address,grade-title,point,employment-start,grade-start,type)
Police-staff(staff-no,uniform-code, description)
Civilian-staff (staff-no, security-level)
Pay(grade,point,amount)
Staff-previous-grade(staff-no,grade,point,start-date,end-date)
Staff-deduction(staff-no,deduction-type,)
Deduction(type, amount)
Police-allowances(staff-no,allowance-code)
Police-overtime(staff-no, month, no-hours, type)
Uniform(uniform-code,uniform-type)
Allowance(code , description, monthly-amount)
Overtime(type,description)
Grade-overtime-rate(grade-title,type,hourly-rate)
Overtime-worked(staff-no,type,month,no-hours)

b) The operation would involve the following algorithm:

```
Check type of employees

If civilian;
    calculate pay based on grade and point less deductions

If police;
    calculate pay based on grade and point less deductions plus
        overtime and allowances
```

Since there is a fair amount of complexity, it is unlikely that the operation can be achieved in standard SQL. An SQL that was extended to include control statements and arithmetic operations might be usable but it is likely that a program would need to be written that included embedded SQL to access the relevant data.

Marks Breakdown

a)	Classes to relations	(2 marks)
	Association	(3 marks)
	Generalisation	(3 marks)
	Aggregation	(2 marks)
b)	Reasonable set of relations	(4 marks)
	Generalisation mapping	(2 marks)
	Association mapping	(2 marks)
	Aggregation mapping	(2 marks)
c)	Algorithm	(3 marks)
	SQL consideration	(2 marks)

QUESTION FIVE

Describe and illustrate the following five diagram types in UML. For each of the diagram types, state its role in the systems design process.

a)	**Use Case Diagram**	**(5 marks)**
b)	**Collaboration Diagram**	**(5 marks)**
c)	**Sequence Diagram**	**(5 marks)**
d)	**State Diagram**	**(5 marks)**
e)	**Activity Diagram**	**(5 marks)**

Candidates generally answered parts a) Use Case Diagram, c) Sequence Diagram and d) State Diagram, fairly well. However, many confused b) Collaboration Diagram with a Class diagram and thus described the wrong diagram.

Additionally, there was widespread confusion between e) Activity Diagram and Critical Path Analysis, which is a project management tool to schedule the various activities within systems development and is not a UML model.

As a consequence of these misunderstandings many candidates were in effect only earning marks on only 60% of the question.

Another common area where candidates lost marks was that many did not include a diagram to illustrate their answers, as requested in the question. Those that did tended to achieve higher marks.

Answer Pointers

The candidate should give a simple illustration of each diagram

a) Use Case Diagram

Use Case Diagram Use case shows the behaviour of the system form the user's perspective. Different users interact with the system in different ways and the perspective of each user can be shown in a use case diagram or model. A use case is drawn as an ellipse, which represents a task. An actor (or user) is associated with the task via a connecting line. The use case diagram combines a number of use cases to show overall interaction from the users' perspective. The diagram consists of matchstick men representing users and nodes representing tasks with connecting lines in between to show which task are performed by which actors. Optionally there can be a box around the set of activity nodes to show the system boundary. An illustration should be given. The main roles played by this diagram in systems design are to plan development and validate system.

b) Collaboration Diagram

A collaboration diagram shows the interaction between objects in the system to perform a task. It may consist of actors, objects, links and control flows. Actors are shown as matchstick men, objects as rectangles labelled ObjectName:ClassName, links are shown like associations in the class model and control flows as numbered labelled arrows where each labelled arrow represents a message sent from one object to another. The role of the collaboration diagram is to record in detail how objects interact to perform a task. It relates to the use case and class diagrams and helps in development and validation.

c) Sequence Diagram

Sequence diagrams show interaction between objects from a different perspective. They show interactions on objects from the perspective of an object's lifetime for a particular task. Objects and actors are shown on the top of vertical dashed lines. Each line represents time from top to bottom and is known as a lifeline for the object or actor it represents. Messages are shown on arrows from the lifeline of one object to that of another. Sequence diagrams serve to show interactions between objects from a time perspective and can be used as a development and validation aid.

d) State Diagram

An object’s decision on what to do upon the receipt of a message depends upon its state, i.e. the values of its attributes at a particular time. The different states that an object can be in and the causes of such states can be shown on a state diagram. The state diagram shows the different valid states of a particular object class in rectangles and the messages that cause the state transitions as labelled arrows. The rectangles contain labels indicating the state of the object. Entry and exit events can also be shown in the rectangles. Filled small circles are used as start and stop markers. State diagrams are useful for development and validation

e) Activity Diagram

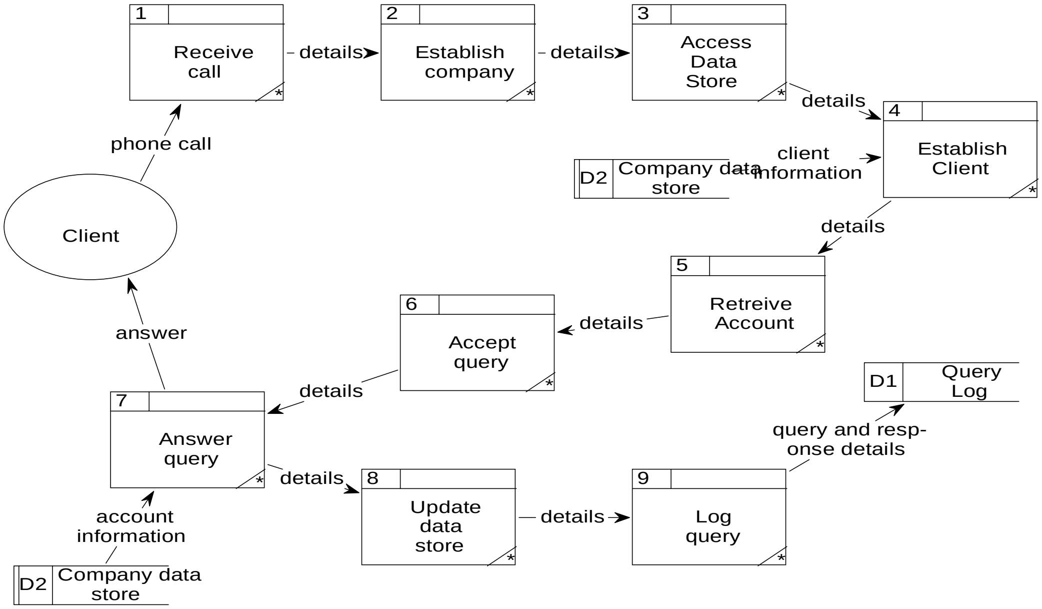
Activity diagrams show how activities are co-ordinated. They show event chains. They are an extension of state diagrams in that they deal with a set of interacting classes rather that a single class and they also have addition control notation. They show dependencies between activities. Activities or states are shown as rounded side rectangles, a transition is shown as an arrow, a horizontal bar is used to indicate synchronization, a decision diamond is used to show decisions. Start and stop markers are used as in state diagrams. . They are useful to show how individual use cases relate to other use cases and can be used to show the functional or work flows within the system. They can be used to show how a process could be implemented.

Marks Breakdown

a)	Description	(4 marks)
	Role	(1 mark)
b)	Description	(4 marks)
	Role	(1 mark)
c)	Description	(4 marks)
	Role	(1 mark)
d)	Description	(4 marks)
	Role	(1 mark)
e)	Description	(4 marks)
	Role	(1 mark)

QUESTION SIX

The following outline data flow diagram describes a business process for an operator at a call centre that handles enquiries for a number of different companies.



a) Produce a structure chart from the diagram (15 marks)

b) State the main principles used to construct a structure chart from a data flow diagram (6 marks)

c) State any difficulties you encountered or which might be encountered when producing a structure chart from a data flow diagram (4 marks)

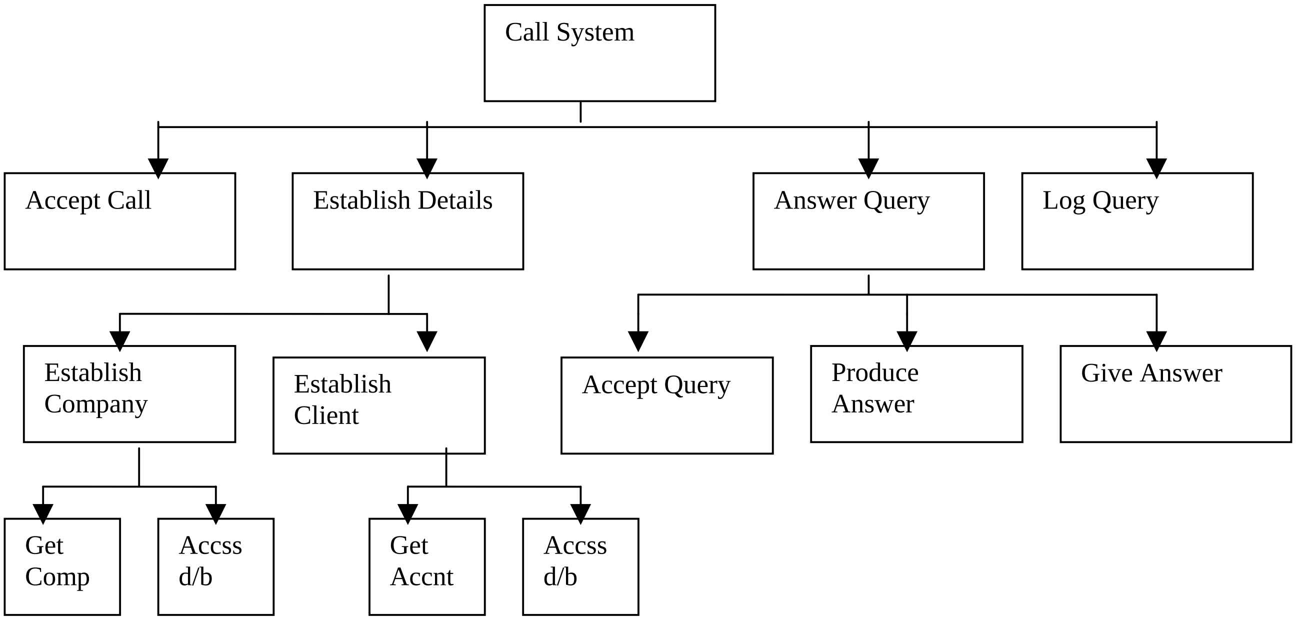
Overall this question was answered fairly well with most candidates producing a reasonable structure chart for part a). Most produced a good mapping of processes to modules though some did not group the modules logically and several failed to include appropriate data couplings. A common mistake was to produce a more or less linear structure chart instead of a 'tree' structure.

In part b) many candidates outlined the principles of producing a structure chart as given in the answer pointers above, however, many discussed the use of transform analysis and transaction analysis in the production of a structure chart. Credit was given to these students depending upon the merit of their answer. Several candidates made statements such as "Identify processes", and then failed to state that these processes then became modules in the structure chart; or "Look at dataflows", and did not go on to explain that these then become data couplings. Candidates are advised to be explicit in answering questions.

A number of candidates ignored part c) of the question and consequently lost marks. Candidates are advised to answer all parts of a question.

Answer Pointers

a) A structure chart such as the following might be produced. Data couples should be shown on the diagram (not shown below because of on-line limitations)



b) The following principles should be followed:

1. Create an overall control module.

2. Each process becomes a module in the structure chart.

3. Each flow becomes a coupling.

4. Add control flags where needed to pass condition information.

5. Group main input and output modules together under a higher level module.

6. Decompose modules that perform more than one function.

7. Group modules that work towards the same identifiable function.

8. Complex flows may be broken down into simple couplings.

c) As a higher level tool, the data flow diagram might not contain enough detail. For instance optionality of process might not be clearly shown in a data flow diagram. Specifics of contents of flows might not be shown. The software designer may need to make assumptions or go back to the analyst who produced the diagram for further details.

Marks Breakdown

a)	Processes to modules	(5 marks)
	Module Groupings and Structure	(5 marks)
	Appropriate Data Couplings	(5 marks)
b)	Each principle	(1 mark up to 6)
c)	Possible difference in perceived levels	(2 marks)
	Example /other points	(2 marks)