THE BCS PROFESSIONAL EXAMINATION
Diploma

April 2005

EXAMINERS' REPORT

Systems Design

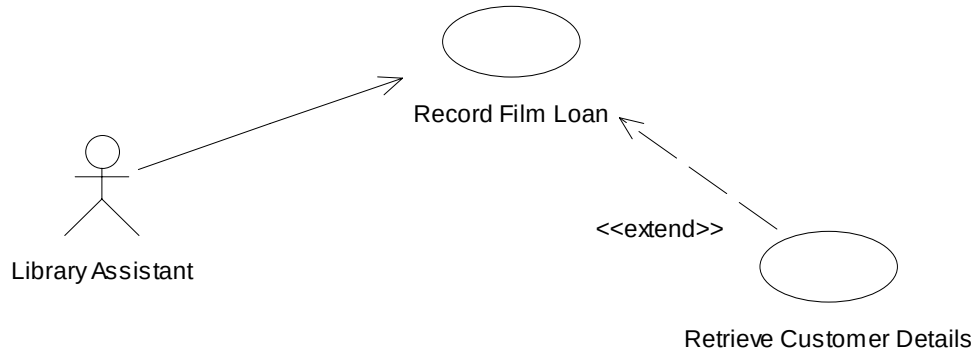
Question 1

1. *a)* Write a *brief* explanation of the purpose of TWO of the following UML diagrams as used in Object Oriented systems design. You should illustrate your answer with a labelled diagram:

i) deployment diagram
ii) package diagram
iii) statechart (state diagram)

(2 x 5 marks)

b) The following fragment of a Use Case diagram and entry from the Use Case Catalogue (**Figure 1** below) is taken from a film library system. Draw a labelled Sequence Diagram for the Use Case *Record Film Loan* to illustrate how objects will interact during the execution of the Use Case. **(15 marks)**



Use Case: *Record Film Loan*
Primary Actor: *Library Assistant*
Goal: *To facilitate the film rental library hiring out films to registered customers*
Preconditions: *Film title is available for loan*
Success Condition: *A specific copy of a specific film title is recorded as loaned to a specific registered customer*
Main Sequence:

1. The library assistant swipes the customer's membership card
2. The screen displays the membership number and confirms the customer's status as being in good standing
3. The library assistant scans the film copy's bar code
4. The system displays the film title, loan type and price and confirms availability for loan
5. The library assistant confirms payment received
6. The system displays the due return date and prints a receipt

Fig. 1

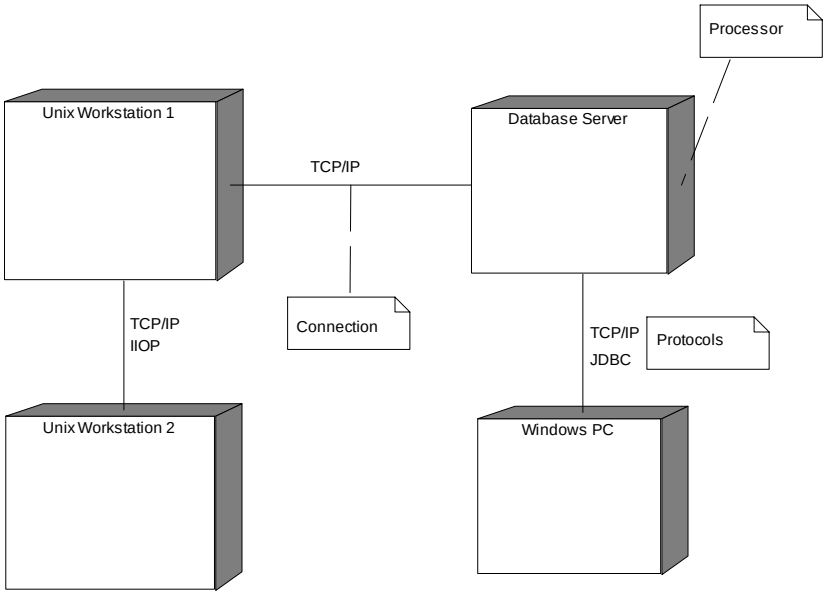
Answer Pointers

1a)

(i) Deployment Diagram:

- shows the physical relationships among software & hardware components
- shows how packages/components are implemented on hardware platforms
- can show particular 'named' machines
- show communication links and protocols between machines

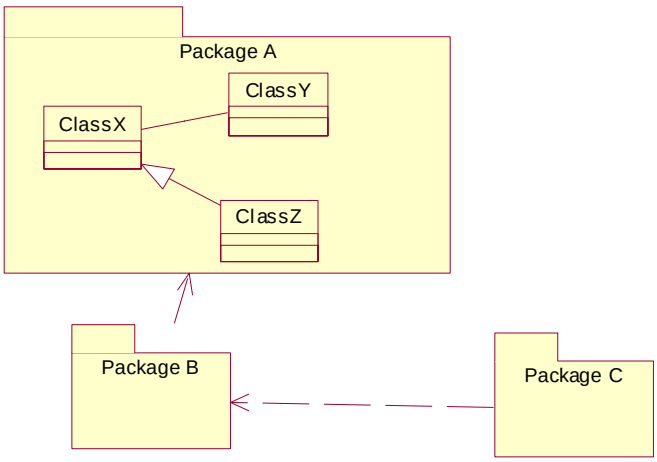
(5 marks)



(ii) Package Diagram

- shows packages (groupings) of classes
- also shows the dependencies among them
- used to show software architecture in a logical way
- packages can be nested within other packages

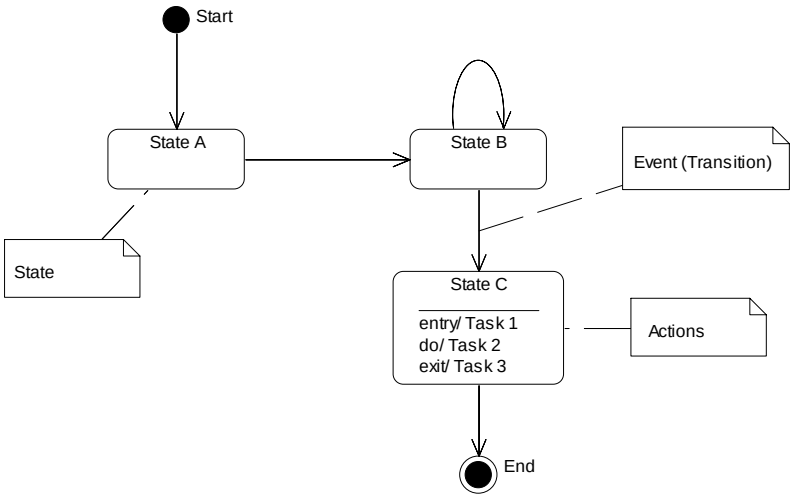
(5 marks)



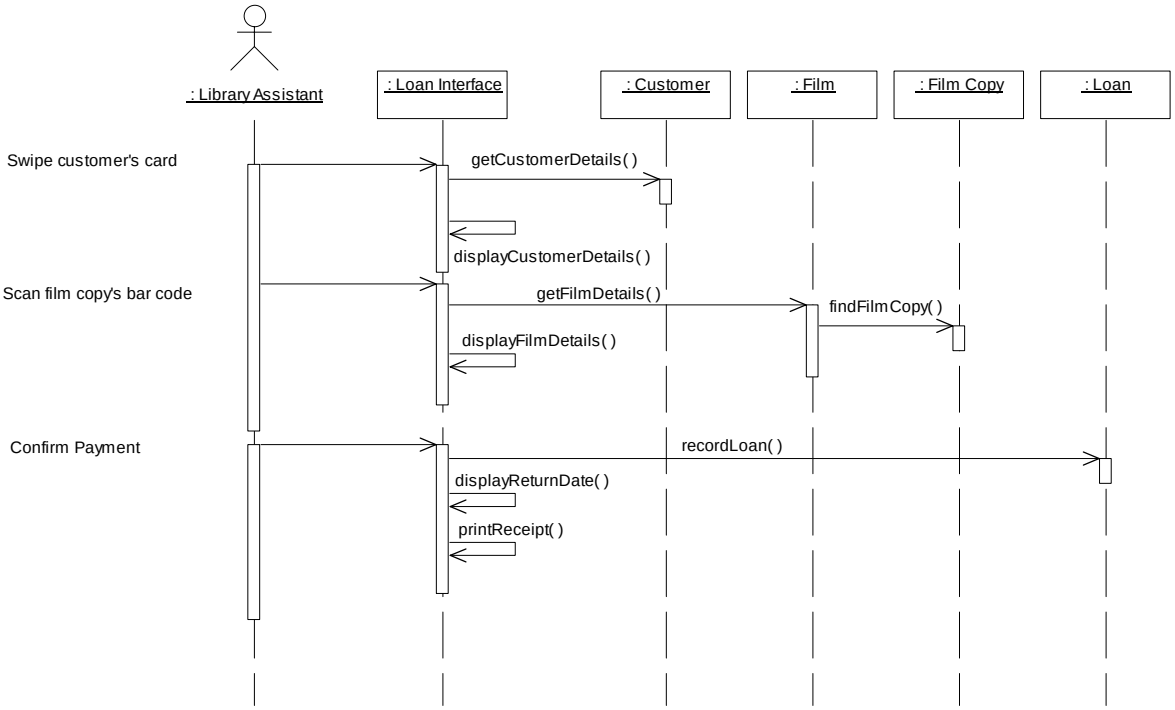
(iii) Statechart Diagram

- dynamic view of the system
- is drawn for any single class of objects with non trivial behaviour
- shows all events affecting that class & all its possible states
- state is determined by the value of an objects attributes and its associations

(5 marks)



1b) Sequence Diagram for the Use Case *Record Film Loan.*



Identification of classes = 5 marks (suitable alternative class names accepted)
Identification of messages = 4 marks (suitable alternative message names accepted)
Logic of interactions= 3 marks
Labelling of diagram = 3 marks

(total 15 marks)

Examiner's Comments

This question was popular with 67% of candidates attempting it. Of those who did it, approximately 64% achieved a pass mark for it and there were some good answers.

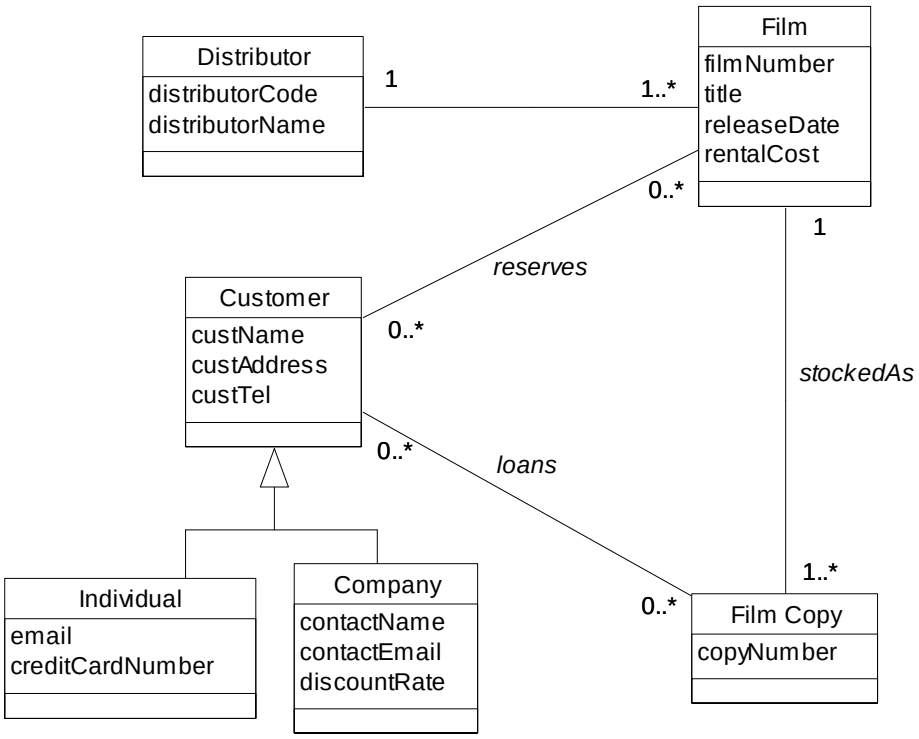
Part (a) was a straight forward demonstration of knowledge with students tending to score fairly highly. Candidates showed much less confusion between UML diagram types than in previous years. However, some candidates replicated answer pointers from previous papers without adding any further detail.

Part (b) required the demonstration of the candidates' ability to apply knowledge and understanding of Sequence Diagrams to a given Use Case and this part was answered less well. Only a small number of candidates were able to identify all of the required classes for the collaboration and to show suitable message passing among them. Nevertheless, most candidates were awarded marks for the classes and message sequencing that they had identified. Generally, the logic of the interactions was confused, with only a small number of answers fully documenting them.

Question 2

2. The following class diagram fragment (**Figure 2**) is part of a design for a film library system. The library not only stocks popular feature films, it also stocks a wide range of training films which are loaned out to companies for their staff development programmes.

The objects represented by the classes in this class diagram are to be stored using a relational database management system (RDBMS).



a) Briefly explain how each of the following elements of the class diagram may be mapped to a RDBMS Schema:

i) a class **(2 marks)**

ii) a one-to-many association **(3 marks)**
iii) a many-to-many association **(4 marks)**
iv) the 'Customer' inheritance hierarchy **(6 marks)**

b) Produce a suitable relational schema (set of normalized tables) from the above class diagram. You *do not* need to show evidence of the normalization process. **(10 marks)**

Answer Pointers

2a)

(i) class

For each class create a relation, determine the primary key (if there are no suitable attributes for this create a primary key attribute e.g. Customer needs a Cust-ID).

(2 marks)

(ii) one-to-many association

For each class create a relation, determine the primary key (if there are no suitable attributes for this create a primary key attribute e.g. Customer needs a Cust-ID). Post the primary key from the relation created for the uni-part class (one end), into the relation created for the multi-part class (many end) to become a foreign key.

(3 marks)

b. many-to-many association

For each class create a relation, determine the primary key (if there are no suitable attributes for this create a primary key attribute e.g. Customer needs a Cust-ID). Create a new relation for the association, create the primary key as the composite of the primary keys of the relations crated for the classes participating in the association.

(4 marks)

c. the 'customer' inheritance hierarchy

Students may identify one of three strategies:

b) implement all classes as tables posting customer PK into child class tables;

c) only implement the superclass; subclasses collapse into attributes (with null values where not used);

d) only implement subclasses; superclass is redundantly copied in each subclass.

(4 marks for any suitable strategy + 2 if existence of >1 strategy is identified, total 6 marks)

(total 15 marks)

2b) The following is a sample solution. Students answer may differ depending upon strategy chosen for dealing with hierarchy – marked on merit.

CUSTOMER (Cust-ID, Cust-Name, Cust-Addr, Cust-Tel)
COMPANY (Cust-ID, Contact-Name, Contact-Email, Discount-Rate)
INDIVIDUAL (Cust-ID, Email, Credit-Card-Number)
DISTRIBUTOR (Distributor-Code, Distributor-Name)
FILM (Film-Number, Title, Release-Date, Rental-Cost, Distributor-Code)
FILM COPY (Copy-Number, Film-Number)
LOAN (Copy-Number, Cust-ID, Loan-Date, Return-Date)
RESERVATION (Film-Number, Cust-ID, Reservation-Date)

Identification of tables = 4 marks (adjusted according to chosen strategies in (a))
Suitable use of PKs & FKs = 3 marks
Assignment of attributes = 3 marks

(total 10 marks)

Examiner's Comments

This question was popular with most candidates (93%) attempting it. Of those who did it, approximately 79% achieved a pass mark for it and there were some very good answers. Other answers could have been improved as some candidates for part (a) replicated answer pointers from previous papers without adding any further detail. This was particularly true of the 'Customer' inheritance hierarchy, where the strategies were stated with little demonstration of understanding nor the significance of the design consequences of each strategy.

A small number of weak candidates did not cover how tables in a RDBMS could be designed to store objects from the classes represented in class diagram, but basically described *what* a class, a 1 to 1..* association, aggregation etc. are; which is not what the question is asking.

Part (b) was fairly well answered; however, some candidates having identified suitable strategies in part (a) did not apply those principles to the production of the normalized tables in (b). Consequently, the LOAN and RESERVATION tables were frequently missed, relationships not properly established through posting Foreign Keys and the strategies for dealing with inheritance and aggregation were ignored.

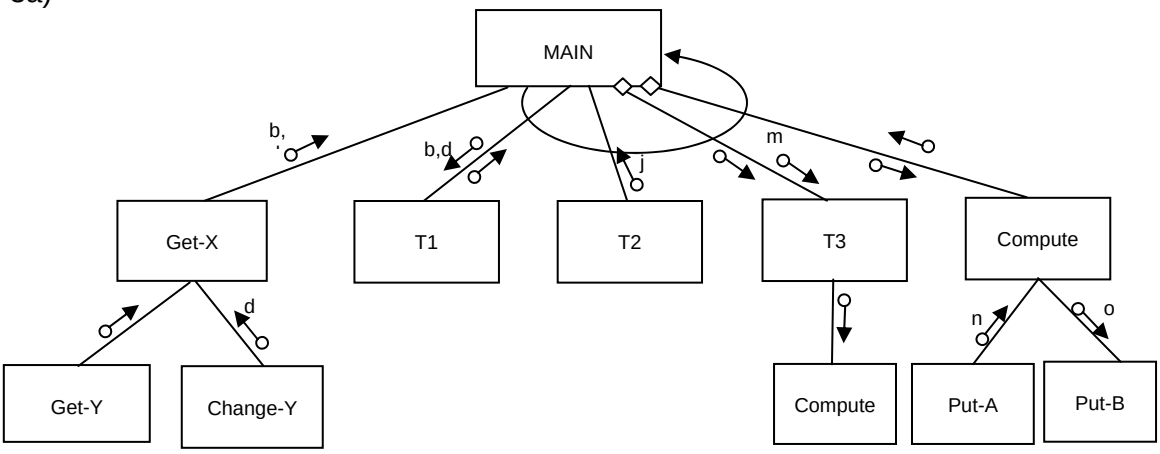
Question 3

3. *a)* With the aid of a labelled diagram, explain how a Structure Chart is used to describe module architectures in a structured programming environment. **(12 marks)**

b) With reference to transform analysis and transaction analysis, discuss the main principles of how a structure chart may be constructed from the information contained in a data flow diagram. **(13 marks)**

Answer Pointers

3a)



A labeled diagram of an example structure chart should be provided and should show the main components that may be included. Marked on merit.

8 marks for diagram showing components such as:

- Program modules (rectangular boxes)
- Main module
- Connections (lines between modules)
- Data couplings (arrows)
- Decision points (diamond symbol)
- Repetition (looping arrow)

4 marks for appropriate description

Total 12 marks

3b) General principles:

Create an overall control module

- Each DFD process becomes a module on the structure chart
- Each flow becomes a coupling
- Control flags added as appropriate to convey condition information
- Group I/O modules under higher level module (transform centres)
- Group modules that work towards same identifiable functions (transaction centres)
- Decompose complex processes & flows

Transform Analysis – uses DFD to identify transform centres (central process with well-defined I/O streams) and data couplings. Processes that provide I/O are converted to modules with additional modules added to allow get/put operations.
Transaction Analysis – uses DFD to identify transaction centres (process splitting input and calling different transactions) and also repetition and decision.

Transform analysis & transaction analysis are the first steps in constructing a structure chart, iterative refinement will be necessary.
(Candidates may support their answer with a diagram to show the structure & syntax).

6 marks for suitable general principles
2 x 3 marks for suitable explanation of transform & transaction analysis

Total 13 marks.

Examiner's Comments

This question was less popular with candidates, with 49% attempting it. Of those who did it, approximately 68% achieved a pass mark for it. This question was generally well answered by most students.

In part (a) most candidates were able to identify the general features of a structure chart and to produce a reasonable diagram in order to aid their explanation. There was little difficulty in identifying modules, main and data couplings, but only a few candidates covered repetition and decision points. A number of candidates failed to discuss how the structure chart is used to model a structured implementation e.g. explaining the significance of data couplings, main module and decision points.

In part (b) most candidates were able to identify the main principles in the development of a structure chart. Several candidates were able to discuss transform analysis and transaction analysis in some detail. However, others merely stated, for example, that transform analysis involves the identification of a transform centre without explaining what a transform centre is.

Question 4

4. Write notes on all of the following, indicating the contribution of each to designing and developing a web-based system.

i) XML **(5 marks)**

ii) CSS (Cascading Style Sheets) **(5 marks)**

iii) Cookies **(5 marks)**

iv) SQL **(5 marks)**

v) Web page frames **(5 marks)**

(5 x 5 marks, total 25 marks)

Answer Pointers

(a) XML –Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML. Originally designed for large-scale electronic publishing. XML is playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere, and provides a widely understood standard.

(b) CSS – Cascading Style Sheets. Allow presentation rules to be summarised so as to be kept consistent & changed easily, through separation of content and presentation. Thus they can help maintain a consistent look and feel for a set of web pages.

A style sheet is made up of style rules that tell a browser how to present a document Style sheets also offer flexibility in terms of the presentation effects that they provide. Properties such as colour, background, margin, border etc. can be applied to all elements. With just HTML, you rely on attributes like BGCOLOR, which are only available for certain elements. Style sheets give the flexibility of applying a style to all paragraphs, or all level-two headings, or all emphasised text for example.

(c) Cookies – A cookie is a variable that is set within the web browser, and can be passed as a parameter in transactions to and from the server. It is also saved locally so can be available for future sessions, depending on its associated expiry date. Some people regard cookies as a threat to privacy.

There are various reasons a given site would wish to use cookies. These range from the ability to personalise information (like on My Yahoo or Excite), or to help with on-line sales/services (like on Amazon or eBay), or simply for the purposes of collecting demographic information (like DoubleClick). Cookies also provide programmers with a quick and convenient means of keeping site content fresh and relevant to the user's interests. The newest servers use cookies to help with back-end interaction as well, which can improve the utility of a site by being able to securely store any personal data that the user has shared with a site (to help with quick logins, for example).

(d) SQL – is an ANSI standard computer language for accessing and manipulating databases, and is used on the server side of a web implementation.
This allows a web page implementation to be (largely) DB independent. Unfortunately, there are many different versions of the SQL language. To comply with the ANSI standard, SQL versions must support the same major keywords in a similar manner (including SELECT, UPDATE, DELETE, INSERT, WHERE).

(e) Web page frames – allow authors to present documents in multiple views, which may be independent windows or sub-windows. Multiple views offer designers a way to keep certain information visible, while other views are scrolled or replaced. For example, within the same window, one frame might display a static banner, a second a navigation menu, and a third the main document that can be scrolled through or

replaced by navigating in the second frame. Frames add structure to a flexible design. But they are controversial, and many people don't like them!

Up to 4 marks each for stating what the term means.

Extra mark for any design awareness, e.g. CSS helping consistency, cookies helping usability.

SQL means don't have to worry about which database.

Examiner's Comments

This was one of the more popular questions, answered by over ¾ of candidates, but answers were disappointing, achieving fairly low marks in most cases. Only about 40% achieved a pass mark. Candidates were probably overconfident; recognising most of the terms, but not being quite sure what they were used for.

a) Several seemed to confuse XML with XHTML, and some appeared to think it was like SQL.
Also answers tended to be rather vague.

b) Very few understood what Cascading Style Sheets were used for.

c) Some regarded cookies as programs or viruses.

d) Most candidates seemed to know what SQL was and how it fitted in.

e) Frames were not well understood or described, though some of the problems with frames were mentioned.

Some candidates forgot to mention developing a web based system in their answers, and in general understanding of the good and bad features of each was rather limited.

Question 5

5. Users may make a variety of mistakes when entering data into an information system.

a) Discuss ways in which the following aspects of User Interface Design can help prevent such errors occurring.
- *i)* Form Design
- *ii)* On-line feedback
- *iii)* Help facilities **(15 marks)**

b) Explain with examples FIVE specific ways fields can be validated in order to prevent or detect errors. **(10 marks)**

Answer Pointers

a)

i) **Form Design**

Meaningful title – so the user knows the purpose of the form
Terminology – use terms that the user is familiar with
Layout – lay out fields logically, don't cram them together
Tabbing order – should follow the structure and layout of the form, probably left to right and top to bottom.
Consistency – of look and feel between fields and forms
Defaults – provide default values where appropriate
Units – make clear the type of data units required for entry
Captions – always provide captions or labels adjacent to fields
Format – provide formatting examples where appropriate

ii) **On-line feedback**

Error message when user enters something invalid or inconsistent, or a set of data that is incomplete
Audible signals (beep) to alert user, e.g. if field full or illegal character ignored
Confirmation screen – summary of information for user to accept
Cursor style – hourglass indicating system is busy and cannot accept new command/input
'Are you sure?' alerts, e.g. when closing screen without saving

iii) **Help facilities**

Well structured; at overview level as well as field level; illustrative examples, details of ranges etc. In fact an on-line manual.
Search facility, for a particular word or phrase
'how to' information,
F1 for context sensitive help

b) Five sensible examples, 2 marks each. The possibilities are numerous, and might include:

- List of options rather than free text data entry
- Only allow valid characters
- Spell checker (can only be advisory)
- Number in range check (may be advisory)
- Slider bar for numeric data (approximate)
- If data legal but unlikely, "are you sure?" message
- Exit without saving – "are you sure?"
- Missed field – display message and highlight
- Summary screen asking for confirmation, e.g. with online ordering
- Provide context sensitive help, e.g. press F1 when in field

Examiner's Comments

This was a very popular question, attempted by more than 80% of candidates, and with reasonable success. Over 70% of these achieved a pass mark. There was however quite a lot of repetition in the answers, and for many "Form Design" simply meant listing the objects making up the form.
They should have written far more about the form as a whole in their answers to this section.

The weakest parts of the answers were to the On-line feedback section, a(ii), and candidates were unable to think of much variety in different forms of feedback.

Candidates were quite aware of Help facilities, a(iii), and their usefulness, possibly because of last year's answer pointers. This section was consequently quite well answered.

Another weak part was describing some of the field validations that could be applied, and the five different examples given were often very similar. A bit more variety and imagination would have helped here, as well as the part about on-line feedback.

Question 6

6. An application needs to show a map of a selected geographical area.

a) Design a screen that displays a map and a variety of controls to show different areas by the following means:

i) Entering latitude and longitude
ii) Panning and zooming
iii) Finding by place name
iv) Going back to a previous view **(16 marks)**

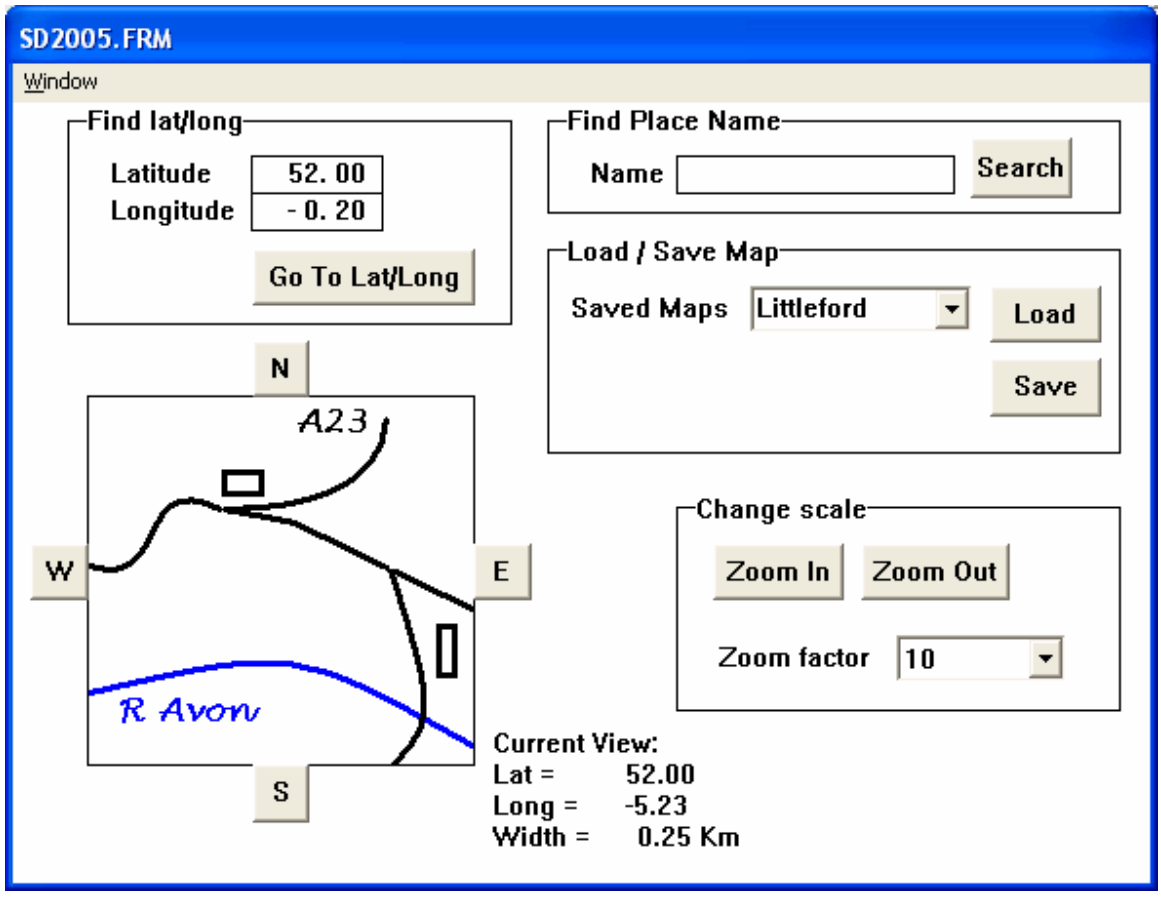
b) Justify the design of each control. **(9 marks)**

Answer Pointers

Marks awarded on merit for quality and completeness of sketch and/or supporting notes.

(a) Two text boxes and a button. Some validation
(b) Pan NSEW and zoom in/out. Display of current centre/range useful
(c) Text with search engine behind it
(d) History or Save List, record the lat/long or grid ref rather than whole map!

Example sketch:



Each control should have an explanation of how it works and why it has been done that way, for example:

Panning is provided by 4 buttons, “N”, “E”, “S”, “W”, positioned at the appropriate sides of the map. These move the map by half its width, so there is some overlap to help the user see what has happened.

Examiner’s Comments
This was the least popular question, and under half the candidates attempted it; but those who did were quite successful; over 70% achieved a pass mark and there were some very good answers.

Most candidates gave a workable sketch of the mapping screen, with some sensible controls, although some seemed to require several different windows to achieve this, and there was disappointingly little explanation. Most candidates thought of returning to a previous view as being just a “back” button, as with Internet Explorer, but some did think of including “save” and “load” buttons.

Many candidates were aware of web-based mapping systems, and this helped them to an extent, but also limited the completeness of their answers.