SYSTEMS ANALYSIS

TUTORS
These notes are designed to assist teachers of the course and are in a condensed format. Teachers should also consult the syllabus for this module and adapt these notes accordingly by using extra examples and by filling out this material with detail. Students will be expected to apply the material in a BUSINESS environment and with reference to the particular situation specified in the examination question. Repeating these or any other notes in a generalised form is unlikely to satisfy the required answers for any question.

Candidates need to have an understanding of

- The need for systems analysis for business in the development of new systems.
- The role of the systems analyst in this development.
- Other people involved – who they are, at what stages in the development and how they are involved.

DEFINITION

System is the grouping of methods and data to provide the required information and outputs in a business. Systems outside of business of public administration need not be considered

Systems Analysis is the process of investigation of a system's operation with a view to changing it to new requirements or improving its current working.

NEED FOR SYSTEMS ANALYSIS

Business systems are usually complex. Making changes to a system without reference to its effects on other subsystems or current working practices could result in a worsening rather than improvement. Systems analysis will identify

1. outputs and processing needed.
2. data required to provide this processing and output.
3. role of people in the process.
4. security aspects to ensure the efficient continuation of the business.
5. costs of providing the system.

THE ROLE OF THE SYSTEMS ANALYST

- Liaison with senior management and the computer manager (Data Processing Manager [DPM]) on the need for new/improved systems. Agree guidelines for the development of a new system and draw up guidelines for an initial investigation and a feasibility study.
- Investigate aspects of the business and produce a Feasibility Report that will be presented to the board or suitable body.
- Attend meetings to decide the way forward and provide technical expertise to the board in making this decision. Recommend a particular solution.
- Make detailed investigations of the affected sections of the business and consult with the users who are using the current system.
- Design all aspects of a new system. Report on developments to the board and identify any likely changes needed/problems discovered as a result of investigations. Produce a systems specification.
- Begin the implementation by convening a meeting of the new system designers. Provide the programmers with individual program specifications. The programmers may actually be database specialists. Set up planning schedules.
- Order Hardware and any structural/electrical wiring changes needed to buildings.
- Liase with programmers on progress. Chase up problems. Begin preliminary user training.
- Supervise testing of new programs. User training with programs.
- Supervise completion of documentation.
- Acceptance tests. Changeover. Hand-over
- Monitoring of existing systems. Regular reviews to see if changes are needed.
- Supervise maintenance and inclusion of new requirements.

INVOLVEMENT OF OTHER STAFF

1. Departmental managers & board. Initiate new systems.
2. Departmental managers. Brief systems analyst on departmental needs, identify key processes, documents and key workers. Support of user staff during the investigations and development. Agree a means of analysing the current system.
3. Users. Assist manager and systems analyst in providing information about current systems including the documentation used and the processes undertaken. Identify particular individual needs of a new system. Undertake training from initial outlines of the new system to specific training on using it.
4. IT Technical staff. Installing and testing new equipment.
5. Development Programmers. Initial briefing about the whole new system, one-to-one briefing about individuals own designated tasks. Keep systems analyst updated on progress and problems. Assist with testing of the new system. Assist with training users. Provide documentation.
6. Maintenance Programmer(s). Become familiar with all programs including one not written by the programmer. Perform changes as required as a priority.

STRUCTURED SYSTEMS ANALYSIS AND DESIGN METHODS (SSADM)

SSADM was designed to formalise the stages of the Systems Life Cycle from planning through to implementation and maintenance.
SSADM looks at a system from three points of view:

1. The processing or functionality of the system – how data moves around a system and changes as it is processed.
2. The structure of the data which is likely to remain more constant than the processes acting upon it.
3. The effect of external events on the data including the passage of time.

SSADM provides sets of standard analysis and design techniques. It separates the logical and physical components of a system. The stages of the systems life cycle are effectively broken down into a series of modules (called **stages**) with standard method of approaching and dealing with them. These are (numbered here as they are normally labelled):

0. Feasibility
1. Investigation of current requirements
2. Business systems options.
3. Requirements Specification.
4. Technical Systems Specification.
5. Logical Design.
6. Physical Design.

SSADM & Computer Assisted Software Engineering (CASE) TOOLS

CASE tools are software tools that enable the systems designer to develop and maintain a new system and provide the appropriate documentation. These tools support the production of E-R diagrams, Data Flow diagrams, structure charts and database design. The tools have become sophisticated so that they can be integrated – one tool can be linked to another. For instance, an E-R diagram can be linked to database design to assist with the structure. Two tools can be used to check the system from two different points of view so that that design can be validated as workable.

SYSTEMS LIFE CYCLE

Overview - This topic has not been answered in sufficient detail in past papers. In particular, the lengthy and expensive implementation stage is often totally ignored or treated as a minor stage. The problems of converting ideas into computer controlled processes is not fully appreciated. For this reason, this topic is covered in some detail here. Although many new systems are now being developed with a database management system, the main elements of the system life cycle will take place in a similar way. The main difference is that the program stage may be shorter with more user involvement as **prototypes** are produced for user comment and improvement.

1. **Initial System Definition**. Consultations between senior management, departmental managers and technical experts such as the systems analyst define a new proposal. This is

then used to provide guidelines for a Feasibility Study that will enable a decision to be made about further development.

2. **Feasibility Study**.

Purpose – To ensure clear objectives are determined before development takes place, administrative and procedural processes are considered, costs are identified (therefore can be controlled) and a clear plan of development is defined.

Terms of Reference - Identifies the people involved including the leader, time-scales, budgets, area under investigation, clear boundaries of the investigation. It will identify the required outcomes such as the cost-benefits expected, comparison with the existing system, alternative solutions and a recommendation for progression. A report will be produced for discussion at high level.

Report – The report will identify the title, the members of the team, dates of the investigation, a review of the existing system with its defects. It will list the reason why the study was undertaken and a proposed solution with alternatives, some of which could be scaled down version or partial solutions. It will identify costs and benefits of the new system, time-scales for implementation and should recommend a particular solution.

Board Decisions – The board or deciding panel can make any one of many different decisions as a result of reading the report, discussing it at a meeting and questioning the system analyst on any matter that would affect the new proposal. Possible decisions (ranging from the ones with least effect on the company to complete acceptance) are:

a) Reject the report outright and continue with the old system
b) Improve the old systems without needing a re-write
c) Postpone the decision until a later date
d) Call for further information or a more detailed report – perhaps a new study
e) Implement only part of the proposal
f) Implement one of the alternative solutions
g) Implement one of the solutions but in a different way than proposed
h) Accept the main proposal and authorise the project to begin

3. **Investigation** – This is a more detailed investigation of the existing system and will involve working in the user departments to find out how the current system operates in practice. The systems analyst will already have found out the main processes within the department and may need to approach the manager for more information and who should be approached to provide it. Seniority and experience may decide this. The information the analyst will require will include:

a) Precise definition of each process
b) Who performs this process
c) What it involves
d) What data is collected
e) How it is collected
f) What data is stored
g) What documentation/forms are used
h) Where the data then is moved

Possible ways to obtain information could be:

a) **Questionnaires -** Can capture from many (possibly remote) sites, from more people and can be used as a starting point for interviews. The aim is to find out about general processes, documents used, document movements, volumes of data, filing systems could be identified at an early stage in the investigation.
However, questionnaires by their nature may need to be standardised for different areas within the business and are only likely to give an outline for further investigation. Returns could be low unless there is managerial pressure, questions may not be given full attention and may represent what should be performed rather than what actually happens. There could be a feeling by staff of outside interference. A previous explanatory meeting with all staff could have eliminated some of the problems. There could be discrepancies between answers from different staff that should be highlighted and investigated at later interviews.

b) **Interview** – This gives direct contact with actual users. It could be one-to-one or in a small group of staff who perform similar functions. Not everyone can be interviewed because it is time-consuming so key people are chosen. Direct contact with users can improve the developer-user relationship. Actual problems experienced by users can be addressed. Staff are told why they have been picked. Terms of the interview are agreed first. A place and time convenient for the user(s) is chosen. Users are asked to bring typical documentation/forms used. A friendly atmosphere is generated with the correct level of language used – not too technical. Direct questions, one at a time and agree answers before moving on. Points raised from questionnaires are raised. The analyst can give some indication of how the new system might work. Questions are invited. A summary is agreed at the end which might be confirmed in writing later. Analyst will ask about procedures, forms, data, where data comes from and where it is then passed, files used, volumes and frequencies. After the interview, the analyst updates charts (DFDs etc). Progress is reported to manager.

c) Documentation analysis **– Forms, documents (invoices etc) are studied. Where possible, these are replicated in the new system to enhance familiarity. Filing systems, ledgers and procedure manuals, where available, are collected.**

d) **Systems Observation** – This could be watching the people performing their task but it is more likely to be watching the paper work – seeing the route it takes and what happens at each step. The idea of volumes of data can be formulated.

4. **Design-** the analyst designs all aspects of the system from the data collected**.** This involves

a) Process design – system flowcharts, data flow diagrams, decision tables
b) Data structures
c) File design including access method and file organisation to be used. Each data item will be defined by name and size, type and use recorded.
d) Input method determination/forms for data capture
e) Validation to be applied to input data
f) Output design – screen and print layouts
g) Modularization – the analyst identifies how the logical design will break down into smaller modules/programs
h) Security of data (and privacy) to be built in
i) Plan for testing and setting target dates. Method of conversion from old to the new system
j) Training identified

A **system specification** is produced and from this, individual **program specifications** for each program process.

5. **Implementation** – This is the process of programming, testing and training. This stage involves:

a) Ordering and installing any new hardware needed
b) Meeting of analyst and programming team for an overview of the whole system
c) Meeting of analyst with individual programmers to discuss and agree targets etc
d) WRITING THE PROGRAMS – the biggest task in this stage.
e) The testing process is defined with a test plan, test data
f) Testing of modules/programs. Testing of linked programs
g) Documentation completed and checked
h) User training
i) User acceptance tests

5. **Review and Maintenance** – These should be treated separately.

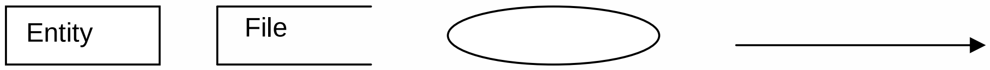
a) Review – Once a system is operational, a close watch is kept on it to identify any discrepancies between expected and actual results. Fixed dates are set for a systematic review of the performance of the system over the first year of operation.

b) Maintenance - Working with the analyst, a maintenance programmer will make urgent changes to bring the system on-line as quickly as possible in the period immediately after the system becomes operational. Further changes will be made in the future as a result of minor changes in business methods, legal requirements and external factors which would change the market position of the company. The maintenance programmer may not be the person who actually wrote the module. This is because the originator may have left or a smaller group of programmers are identified to handle the whole system while others move onto new projects.

RECORDING DATA

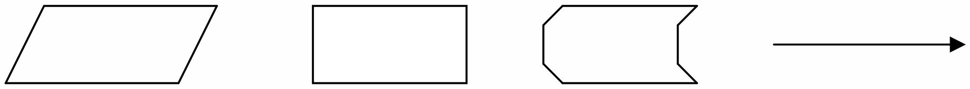
Candidates must be able to interpret and construct each of the following diagrams.

a) **Data Flow Diagram** – This should be drawn with standard symbols – EACH LABELLED.



External entity	Filename	Process	Data Flow
e.g. Customer	e.g. Stock File	e.g. Produce invoices	e.g. Customer order

b) **System Flowchart** – Again standard symbols should be used. This must not be confused with program flowchart that identifies the various steps <u>within</u> a program.



Input/Output	Process	File Access	To next event
e.g. Customer record	e.g. Invoices	e.g. Customer Record	

c) **Decision Tables** – These should always have the four main sections clearly marked.

Decision stub	Decision Entry
Action stub	Action entry

The decision stub entries must be precise. e.g. Age < 16 and Age > 16 as two entries means there is no category for Age = 16. In the Decision entry, columns are drawn with Y or N entries (YES/NO). Blanks or the "–" symbol can be used where this is irrelevant. A blanket cover of all possible Y/N situations should be carefully checked to avoid inconsistencies.

e.g. In one column, it is logically impossible to have a "Y" for Age >= 25 and a "N" for Age >= 16.

The Action entries should each list ONE possible outcome. e.g. reject customer order OR discount = 5%. It is permitted to use "set discount" and list in the Action entries, the actual discount rate attributed to these conditions. So, different values would be in each column or a "–" to show discount is not applicable.

d) **Progress charts/Gantt Charts** - These might take the format of a bar-chart.

```
   MONTH        1   2   3   4   5   6   7   8   9   10
Feasibility     xxxxxxxxx
Analysis                xxxxxxxxxxxxxx
Design                      xxxxxxxxxxxxxxxxxx
Order Hardware                          xxxxxxxxxxxxxxxx
    etc
```

This is a planning chart and can be over-marked/underlined to show actual progress.

DATA CAPTURE

Extensive notes are included in Information Processing notes including how to decide which method would be most suitable for a given application.

FORM DESIGN (PAPER)

Candidates are expected to know what makes a good or bad form. Factors to consider in designing a form for a new data capture situation are:

- Purpose of the form – needs a number/title to identify and distinguish it from others.
- Colour – different colours enable immediate differentiation between different situations. e.g. full-time employees claim travel expenses on pink forms, casual workers use green.
- Size of paper – defined by amount of data to be collected
- Double sided – for larger amounts of data and avoid a second page which could become detached.
- Quality of paper – depends on the conditions of use. A delivery person would need strong paper because of the possible rough usage – completion when standing up. Internal notes could be passed on cheap, thinner paper.
- Order of data – It would assist computer operators if the data on the form is in the <u>same order</u> as keyboard input at a later date. Data on the form which is NOT input should be separate and perhaps at the end.

- Help – some indication of type of data or format. e.g. date of birth 23/6/2008 or number of decimal places needed. Boxes on the form could clearly restrict the entry.
- Serial number – where each form is unique or the number is an identifier for queries. Missing numbers in a sequence then become obvious. Needed where queues are involved.
- Space per item – should be adequate. Excessive space encourages unnecessary information to be entered.
- Font and size – choice could assist people with impaired sight.
- Orientation – landscape or portrait – dictated by the type of data.
- Official use – space provided for staff to mark/initial/date the progress of the form as it is processed.
- Is there an existing form that could be used?
- Where to send the form. Supplying this information separately does not help the sender if the two items become separated.
- Are multiple copies needed? – E.g. Filling in details of an order could have carbon copies for picking list in the warehouse or a delivery note for a driver.
- Cost – is usually not an issue if the form is needed.

SCREEN FORMS

Where data is keyed directly in, perhaps during a telephone conversation, screen forms are needed. The operator is under greater time pressure than the paper form filler. He/she is less likely to make mistakes if the form is easy to use. Factors to assist this are:

- Double space – avoids data being entered on the wrong line
- Colours – to highlight different parts of the form
- Tabulation in aligned columns means particular items are easier to find
- Fixed size boxes assist fixed-sized data items such as reference numbers highlights possible errors at the time of input
- Help should be available but should not restrict the form itself – Function key access.
- Data should be in the same order as the original paper form.
- Place computers in positions where reflections of windows/lights etc will not distract.

OTHER SCREEN OUTPUTS

A limitation of screen output is that it is substantially smaller than the equivalent paper. A person is able to assimilate information on 3 or 4 sheets of papers in front of him/her. This could be in excess of 150 lines compared with perhaps under 30 on a screen. This means that multiple screens will be needed to display large data output. Scrolling is not always satisfactory because mouse control on the scroll bars is not precise. Paging keys might be more useful to enable fast and exact repositioning to occur.

A factor often overlooked is that looking down is still a natural way to read a book or look at sheets of paper. A vertical screen is unnatural for humans to read.

ERRORS

Errors occurring on data entry result in time loss for re-entry and the consequent expense of making the correction. It may be necessary to correct ONE error by making SEVERAL corrections if the one entry has resulted in the data being used in different situations. Errors give a bad image of the company to customers.

CODING DATA

This should not be confused with encryption for secrecy purposes. Candidate answers frequently describe encryption despite the questions clearly excluding this aspect. Data is coded for the following reasons:

1. Codes can be made into fixed sizes – errors easier to spot as a result and more suited to computer storage and fast retrieval.
2. Codes are unique avoiding confusion. Different codes would be used for different sizes/colours of the same product.
3. Codes can be multipurpose in that their component parts help to define different aspects of the item coded.
4. Validation performed by computer enables errors to be reported at the time of input and then corrected. Self-checking Codes – see Validation in Notes on Information Processing regarding check digit systems used for reference numbers.

DATA CODING SYSTEMS

1. **Sequence** – consecutive numbers are used to give priority and to draw attention to missing numbers.
2. **Block** – Groups of numbers are reserved for particular types of items. As a result, there will be gaps in the coding system where some codes are yet to be used. e.g. 1-199 used for employees in section A, 200-399 for section B etc.
3. **Facetted** – Different parts of the code define aspects of the item. e.g. An engineering company that makes screws could use a code such as R1208C for a Round-headed screw 12 cms long with gauge 8 and made of copper. 08 is used for gauge here to allow gauge sizes bigger than 9.
4. **Hierarchical** – A supreme example of this is the Dewey system for cataloguing books. Social Science books are all within the 300-399 section. A subset is education (7) and of this, a subset of education is primary education for the very young (2). A book on primary education would therefore have the code 372.xxx where xxx are further subsections of primary education such as basic number work.

Candidates could be asked to design a realistic and efficient code for a simple defined situation.

OUTPUT DESIGN

Candidates should be familiar with standard formats such as invoices and account statements. In the examination, the candidate could be asked to design a paper/screen output. The answer will NOT be marked on the neatness of the answer but on the shape and workability of finished format – in practice it would clearly be printed and not hand-written. Some past answers have been totally impractical for use in the real business world missing vital information such as titles, invoice totals, VAT/discount, where to send….

NOTE: Candidates could be asked to identify outputs from a particular application. Expected answers might be invoice, account statement, receipt etc. Past exam answers often listed only SINGLE FIELDS such as customer number.

FILES

Extensive notes within Notes for Information Processing cover the types of filing systems available in computer systems. Candidates must be able to:

1. Name the files that would be present in typical business situations. e.g. A retail ordering system would include Customer Identification, Order, Stock, Supplier files.
2. List fields that would be present on each file including the data types (numeric/text etc) and length. Realistic answers are expected. e.g. A person's address would actually be held as several fields in order to enable address printing for use in the mail and to be used for statistical purposes such as identifying how many customers come from a particular town or region.
3. Identify the file organisation method of access needed for a particular application – serial, sequential, indexed-sequential, random.
   Recognise that Indexed-sequential organisation permits multiple uses – sequential access of the whole file (e.g. printing payslips) and direct access to single records (e.g. altering an employee's address or status in the company).

SOFTWARE PRODUCTION

Programs can be generated for a new application by a variety of means.

1. Writing code in-house in the traditional way using a traditional language. This is now becoming a minority way.
2. Designing programs in-house using a database management system. Prototypes can be developed quickly and tested for user comments and amendments. The user is in the forefront of development from an early stage and avoids the situation where the finished program is not exactly what was needed – the user is not as well trained in defining his/her own needs precisely as is the programmer.
3. Purchasing standard packages from a software supplier. It is likely that when the packages are installed, the user will need to customise them to his/her own use. At the very least, this would be to identify the particular printer being used.

4. Contracting a software house to write programs. This is complicated by the need to define the system – this could be performed in-house or also by the software house. With the advent of database systems, a software house can be a one-man programmer.

IMPLEMENTATION

There are many aspects of implementation.

1. Writing/obtaining the programs.
2. Testing
3. Conversion of old files
4. Installing programs
5. Ordering and fitting hardware
6. Documentation
7. Training

TESTING

1. **Test Strategy** – overall method of approach. Modules could be tested individually and then the linked. Which modules should be tested first?
2. **Test Plan** – list of actual tests that will be carried out in sequence with more detail than strategic plan.
3. **Test Data** – list of data to be used. Each set should be selected to test a particular situation (e.g. eligibility for the highest discount rate) - generally only tested once but there may need to be some bulk data to test for bulk situations. e.g. customer orders sufficient items to require a second page on the invoice and hence requiring carried forward totals. Test data must be vigorous enough to ensure faulty input is detected and reported.
4. **Test Log** – after every test, a record should be kept showing the outcomes. This will identify which aspects appear to be correct and which are faulty. An indication of a possible correction should be given.

A fuller treatment of testing can be found in Notes for Programming.

TRAINING

The systems analyst is responsible for training users. The training process could begin at an early stage during the development of the new system.

1. The analyst addresses a meeting of all members of a department affected by the new proposal. This will outline the reasons for the change, the effect on the users, time-scales predicted, training programme details and will give the users the opportunity to asks questions. Particular problem areas in the present system could be highlighted at this point.
2. When the analyst interviews some of the staff during the investigation stage, some more details of the change will emerge.
3. Users could be asked to supply live data for testing and be asked to oversee the results.
4. Users are given written instructions on new procedures.
5. General training on standard features of computer usage – operating system, printers, networks.
6. Formal hands-on training occurs using the new programs.
7. Users are instructed in aspects of security and privacy.

NOTE: Examination questions could ask WHAT training is needed or HOW it could be delivered or WHEN it should be given. Past examination answers have frequently answered the wrong question. The words WHAT, HOW or WHY are likely to be highlighted but this has not improved the response.

SECURITY AND PRIVACY

Security training should cover:

- Proper treatment of hardware.
- Possible hazards in using hardware – trailing wires, food and drink near hardware, continuous use of computers without a break, screen glare etc
- How to deal with and report a problem.
- Company policy on private use of computers – email and internet usage, restrictions on the use of email within the company.
- Backup security facilities. Testing that backup copies actually work.

Privacy training should cover:

- Turning off of unattended machines.
- Use of passwords, including changing passwords.
- Which information is private and the consequences of that.
- The “need to know” policy.

More detail on implementing security is given in Notes for Information Processing.

DOCUMENTATION

1. **System Specification** – This is an overview of the whole system and would outline:
   a) Description of the purpose of the system including the boundaries and limitations
   b) Structure of how the system is broken into modules
   c) Brief description of each module and the inter-linking
   d) System flowchart
   e) File structures including file organisation and file access methods
   f) Outline of input data
   g) Outline of the outputs produced
   h) Security and privacy aspects
   i) Time scales and target dates
2. **Program Specifications** – Outline for the programmer to plan and develop the program
   a) Outline description of the program including limitations
   b) Where the program links with other modules/programs
   c) Files used – detailed on every field.
   d) Input data structure
   e) Outputs produced – printer and screen
   f) Validation to be included
   g) Description of the processes involved
   h) Security/privacy features to be included in this program
   i) Target dates
   j) Testing procedures
   k) Test data – not always supplied
3. **User Manuals** – to assist the user with any problems that might arise
   a) Installation instructions
   b) Running instructions
   c) Options available
   d) Data formats for input
   e) Output details
   f) Help facilities built into the program
   g) Manual procedures – where data is collected from and where it is to be passed on.
4. **Maintenance Instructions** – designed for use by the maintenance programmer.
   a) Systems specification
   b) Program specification
   c) User Manual
   d) Algorithms producing during programming
   e) Listings
   f) Variables list (as used in programs)
   g) Special methods. Perhaps particular data structures have been included to solve some aspect of the problem or a formula devised to calculate one of the outputs.
   h) Test data used.
   i) Test logs to show previous testing.

CHARACTERISTICS OF COMMON PROCESSES

1. **File Sort** – Producing a new file with the same data but ordered on a particular key field in each record. Files are normally too big to read into memory, sort and then output.
   a) Instead, perhaps 10 records are read into memory, sorted and output to file A. Another 10 are sorted and output to file B. A third group are sorted and output back onto A. This alternate process continues until the end of the original file is met.

b) Files A and B are then read in parallel from the beginning. One record of each is compared and the lower key value record (X) (for ascending sort) is output to file C. The file from which this record X is read to replace the record in memory. The two in memory are compared and the process repeated until all records in the two 10-record sets are output. The result is that file C now has 20 records in order. In turn, 20 more records are read in from the files A and B and the ordered output sent to a new file D. Continue this, alternating the files C and D for the output.

c) The process of b) repeats with files C and D now as input outputting to two other files. File spaces A and B are no longer needed and can be reused for this purpose. The result is that blocks of 40 records are in order on each of A and B.

d) The process continues alternating A and B with C and D for input/output files. At each stage, the number of grouped records on each files doubles. This repeats until all the records are in order on one file. The program reports which file contains the sorted data.

2. **File Merge** – Two (or more files) from different sources, already sorted into the same key order, can be combined as one file.

   Open the two source files (A and B). Read one record from each and send the one with the lower key (X) to an output file. Read another record from file X to replace it. Repeat the process until one of the files has been read and transferred to the output file. Copy the remaining records across from the other file to the output file and close all files.

3. **Search a file** – to find records which meet a given condition. The condition details must first be input (e.g. CustomerBalance>=100 AND TodayDate>OrderDate + 30 = customer owing £100 or more and has not paid with 30 days of the order).

   Open the file and read each record in turn checking CustomerBalance with 100 and TodayDate with OrderDate. If the record meets the condition, it is output as a list to the screen or printer OR could be output as a reminder letter fully addressed from the customer record. This repeats until all records on the file are read.

CHARACTERISTICS OF COMMON APPLICATIONS

1. **Stock Control** – This is the process of ensuring that adequate but not excess products are in stock to meet customer demand. It is similar to 3 immediately above. The reordering process involves:

   a) The entire Product file is read, one record at a time and checked for reorder conditions. Check if StockLevel is below the StockMinimumLevel for that product. This should take into account the value of the OnOrder field which shows the number already ordered. If StockLevel + OnOrder >= StockMinimumLevel, reordering is not needed yet.

   b) The number to order depends on the product.

      i. Often, this must be in bulk and in fixed quantities set by the supplier (e.g. food items for a supermarket). Order sufficient to take StockLevel up to but not exceeding StockMaximumLevel.

      ii. Order StockMinimumLevel – StockLevel where fixed order quantities are NOT necessary such as for larger items (e.g. ONE set of a table and chairs).

      iii. Economic Order Quantity – See Notes for Number and Logic.

      iv. Seasonal differences – the stock control clerk would have the chance to override the normal ordering process.

      v. Reordering can either be triggered as a product StockLevel falls below its StockMinimumLevel with automatic orders sent directly to the supplier by **electronic information interchange** OR it can be a regular event (daily/weekly) where all product levels are checked in one stage and reorder lists produced for staff to check before placing orders.

   c) The stock control system must have features to allow other changes:

      i. Add a new product to the stock list.

      ii. Update StockLevel when goods are received.

      iii. As sales are made, StockLevel should ideally decrease automatically

      iv. Write-off goods in stock when are damaged or beyond of sell-by-date.

      v. Remove a product from the stock list so that further ordering does not take place.

      vi. Altering prices including for short-term promotional sales. Products could also be linked so that in buying 3 items, only 2 are charged to the customer OR two different products are priced lower if bought together.

vii. Backup systems for records

2. **Retail Systems** – Retailers buy in products from wholesalers, usually in bulk, and sell individual items to their customers. The features of the system will include:
   a) **Stock Control** – as outlined above.
   b) **Ordering from suppliers** – Main processes are:
   Placing orders, arranging delivery dates, chasing goods expected but not yet received, checking goods received against a delivery note, reporting shortages or damaged goods, checking invoices against delivery, paying the invoice by cheque or through automatic banking systems on-line.
   c) **Selling to customers** – This is effectively the reverse of ordering from suppliers in that the customer now places the order but clearly in smaller quantities (often 1). A distinction needs to be made between
      i. purchases made in a shop where the customer may pay by cash/cheque or credit card
      ii. purchases made in a shop where the customer holds an account and pays the account balance monthly when an account statement is received.
      iii. purchases made by mail/telephone/on-line where the customer pays by debit/credit card
      iv. purchases made by mail/telephone/on-line where the customer pays monthly when an account statement is received.

   The main processes vary according to which of these four possibilities occur. In general the processes will involve:
      i. Recording the purchase and adjusting stock levels.
      ii. Invoicing for a particular order made with a pay-by date.
      iii. Sending monthly statements to account customers.
      iv. Recording payments received where this is physical payment.
      v. Checking at intervals that outstanding bills/statements have been received and if not sending reminder letters.
      vi. Accepting new customers as account holders and checking their credit record.
      vii. Close customer accounts on request or because of payment failures.
      viii. Dealing with customer queries about delivery and failure to deliver.
      ix. Processing credit notes issued as a result of an irregular previous sale.
      x. Backup systems for records

3. **Staff records and Payroll** – These are dual processes and although related, are quite separate. Both aspects would have privacy implications and there should be safeguards built-in to prevent unauthorised access.
   a) **Human Resources**- involves the personnel aspects and will include:
      i. Recording personal details of employees
      ii. Changing details of employees – change of address, promotions etc.
      iii. Check dates for annual review and record new outcomes
      iv. Record job history
      v. Record medical/health history
      vi. Enquiry facilities for authorised users.
      vii. Backup systems for records
   b) **Payroll** – among the tasks are:
      i. Record salary details and job status
      ii. Record time and overtime worked
      iii. Record holiday times due
      iv. Calculate pay due, tax, other deductions and print payslips
      v. Print payroll details
      vi. Handle loans to employees
      vii. Print payments due to other bodies (tax to government, pension contributions to pension supplier etc)
      viii. Produce annual statements of total earning, tax etc
      ix. Transmit authority to banks to make payment transfers
      x. Backup systems for records

4. **Medical records** – at a health centre, doctor's practice or hospital. Privacy issues are involved here as for 3 above. Main tasks are:
   i. Recording patient personal details
   ii. Recording patient medical conditions and history
   iii. Booking systems for appointments
   iv. Checking past patient history at a consultation and recording new treatments
   v. Enquiry facilities for authorised users.
   vi. Handling charges where appropriate
   vii. Backup systems for records
   viii. Stock control where drugs etc are issued
   ix. Staff rotas to man the clinics, wards

5. **Library Administration** – Involves the control of library materials and the borrowing process. Libraries now deal with a range of items beyond large book collections. Main tasks are:
   i. Recording details of new books
   ii. Recording borrower identification details
   iii. Recording the issue and return of borrowed items
   iv. Dealing with charges for borrowing and fines for late returns
   v. Dealing for borrower reservations of items currently out of the library
   vi. Dealing with the transfer of library items which might circulate to and from other associated libraries
   vii. Budgeting for the coming year with regards to stock and staff

OTHER ISSUES REGARDING APPLICATIONS

Candidates must have an appreciation and be able to explain or describe:

1. Files needed in a given application including contents, organisation and access methods
2. Data input and output
3. People involved
4. Manual processes not performed by computer systems
5. Security and privacy of data required

TEACHING APPLICATIONS

Answers to examination questions should be based on the real world and not some idealistic set of steps frequently found in books. An example of a poor answer is summarised here:

Payroll – Refers only to calculating pay from Hours x RateOfPay with perhaps tax deducted at a fixed standard rate and then printing a payslip. This omits all the complexities of payroll such as maintaining of records, accumulation totals, deductions other than tax, how the money is actually paid to the employees and to outside bodies such as a tax office. In reality, there are hundreds of possible ways a pay total can be adjusted from the simple case noted above including subscriptions to clubs, health organisations, pensions, loans, overtime, corrections through changes of conditions…..

A study of a some applications could be undertaken by groups of students, each investigating a different application and reporting back to the whole class in the form of a presentation. Other members of the class should be able to question and offer other information of which they may have personal experience.