

**The Institute of Commercial Management**  
The Fusee, 20A Bargates, Christchurch, Dorset, BH23 1QL, England

**COURSE HANDBOOK**  
**2009**



**Professional Computing & Information Processing**

**Certificate, Diploma,  
and Advanced Diploma**

Institute of Commercial Management, The Fusee, 20A Bargates, Christchurch, BH23 1QL England  
Tel: +44 (0) 1202 490555 Fax: +44 (0) 1202 490666  
Email: [icm@icm.ac.uk](mailto:icm@icm.ac.uk) Web: [www.icm.ac.uk](http://www.icm.ac.uk)



# Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>The Institute of Commercial Management</b>	<b>1</b>
<b>3</b>	<b>ICM Articulation Agreements with Universities</b>	<b>2</b>
<b>4</b>	<b>Learning, Teaching and Assessment Strategy</b>	<b>3</b>
4.1	Assessment	3
4.2	Student Work Load	3
<b>5</b>	<b>ICM Qualifications and Progression</b>	<b>4</b>
5.1	ICM Certificates	4
5.2	ICM Diplomas	5
5.3	ICM Advanced Diplomas	5
5.4	Graduate Diplomas	6
5.5	Relative Positioning of ICM Qualifications	6
<b>6</b>	<b>ICM Academic Calendar</b>	<b>7</b>
6.1	Examinations	7
6.2	Examination Closing Dates	8
6.3	Examination Results	8
6.4	Subject Exemptions	8
6.5	Study Methods	8
<b>7</b>	<b>Copies of past examination papers and other forms of assessment</b>	<b>8</b>
<b>8</b>	<b>Subject advice, educational guidance and student support</b>	<b>8</b>
<b>9</b>	<b>Unit Information</b>	<b>8</b>
9.1	Programme Objectives	9
9.2	Recommended Course Duration	9
9.3	Continuing Education	9
9.4	Entry Requirements	10
<b>10</b>	<b>Course Structure</b>	<b>11</b>
10.1	Unit Syllabuses	11
10.1.1	General Purpose Software	12
10.1.2	Information Processing	16
10.1.3	Programming	21
10.1.4	Numeracy & Statistics	27
10.1.5	Number and Logic	30
10.1.6	Systems Analysis	34
10.1.7	Applications	38
10.1.8	Programming Project	43
10.1.9	Hardware and Operating Systems	49
10.1.10	Managing Computer Systems	55
10.1.11	Networks	59
10.1.12	Feasibility Study	62

---



## **1. Introduction**

---

The purpose of this course handbook is to provide you with information about your programme of study and to direct you to other general information about studying with the Institute of Commercial Management (ICM).

This handbook must be read in conjunction with other guidelines available on the ICM website which will be updated from time to time. The early pages in this handbook provide general information about the ICM and its programmes. Later sections give details of the Certificate, Diploma and Advanced Diploma in Professional Computing and Information Processing programme.

The material in this handbook is as accurate as possible at the date of production and should you have any comments on improvements to this handbook please put them in writing along with the name of the course handbook to [info@icm.ac.uk](mailto:info@icm.ac.uk)

## **2. The Institute of Commercial Management**

---

The Institute of Commercial Management (ICM) has provided support and guidance for life long learning in 120 countries over the past 30 years. The ICM offers awards at Certificate, Diploma, Advanced Diploma and Graduate Diploma levels in a wide range of subject areas.

The ICM is recognised by the UK government's accreditation organisation QCA, as an awarding body that can offer programmes up to doctorate level. Delivery of ICM courses currently takes place in over 600 teaching centres and this provides an extensive global network in which to deliver excellent vocationally-focussed programmes.

ICM is committed to providing high quality global education, training and consulting services which raise performance standards for business, encourage lifetime learning and provide professional status for individuals.

Working in close partnership with commerce, government, education providers, international development agencies and its own professional membership, ICM has built an enviable reputation for excellence based on global standards and integrity.

ICM is known for the design and development of progressive, practical and multi-functional programmes of study for use by universities, business schools, colleges and other education and training providers and is acknowledged as a specialist in this field.

Growth of ICM programmes has been entirely organic in the truest sense and has been achieved through product quality, consistency and tenacity in even the most challenging markets.

### **3. ICM Articulation Agreements with Universities**

---

Recognised as an international examining board for professional business and management studies, ICM is renowned for examining and certifying candidates to an internationally consistent standard.

Qualifications are offered in a number of subjects ranging from accounting and finance to tourism and development, and from marketing and advertising to project management and export.

ICM has formal partnership arrangements with a number of Universities including Birmingham City, Southampton Solent, Portsmouth, Lincoln, York & St John and Sunderland.

Established articulation arrangements provide direct access to the second and final years of a wide range of BA/BSc degrees for ICM Diploma and Advanced Diploma holders and access to Masters programmes for those with Graduate Diplomas. In addition, ICM Certificate holders are eligible for entry to level 1 of appropriate degrees.

Subject to status and grades obtained, students holding ICM professional level Certificate, Diploma, Advanced Diploma and Graduate Diploma awards are accepted onto undergraduate and post-graduate degree programmes offered by institutions in Europe, North America, South East Asia, Sub Saharan Africa and Australasia.

Our strategic partners have agreed that students can enter their degree programmes with advanced standing if they have the right grade profile and are able to meet the other admissions criteria such as language competence.

#### **Entry to the first year**

Application for entry to the first year of a relevant degree programme will be considered for students who have passed the Institute of Commercial Management (ICM) Certificate. Grade requirements are subject to local variation.

#### **Entry to the second year**

Application for entry to the second year of an Honours degree programme will be considered for students who have passed the two year Institute of Commercial Management (ICM) Diploma and have met the following requirements:

- obtained at least 24 Points from 8 subjects leading to the Diploma
- obtained at least Grade C in all subjects in the Diploma

#### **Entry to the Final year**

Students who have passed the ICM Advanced Diploma will be considered for final year entry if they have achieved the following:

- obtained at least 32 Points from 12 subjects leading to the Advanced Diploma
- obtained at least Grade C in all subjects in the Advanced Diploma

The following rules apply in relation to converting ICM subject grades into points:

- an 'A' (Distinction) is awarded 6 points
- a 'B' (Credit) awarded 4 points
- a 'C' (Pass) is awarded 2 points
- a 'D' (Marginal Pass) is awarded 0 points

Students wishing to progress to degree programmes should review the range of courses available at our partner Universities. Information on appropriate University courses can be obtained from our website on [www.icm.ac.uk](http://www.icm.ac.uk).

The Institute's awards are also recognised by leading professional examining boards for either subject exemption or registration purposes. Details of this can be obtained from the appropriate examination boards.

#### **4. Learning, Teaching and Assessment Strategy**

---

Learning takes place on a number of levels through lectures, seminars, class discussion including problem review and analysis.

Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class.

Students are actively encouraged to form study groups to discuss course material which fosters a greater depth of learning experience.

##### **4.1 Assessment**

In virtually all cases, the assessment of ICM subjects is based on a final exam. The ICM has a traditional approach to assessment. Formative assessment is provided by the delivering centres as and when appropriate and may include: Practical IT exercises, in-class tests, role play exercises, presentations and projects. However, in terms of gaining a professional qualification, and in order to maintain standards, the ICM believes that students should be formally assessed on the body of knowledge covered during the course and be examined as individuals on the required learning outcomes.

##### **4.2 Student Work Load**

In accordance with nationally accepted codes of practice in the UK, each 20 credit unit represents a total of approximately 200 hours work.

Typically, an ICM Advanced Diploma comprises 240 Credits being 12 subjects of 20 credits each. Students must complete 4 subjects at each level leading to the awards of Certificate, Diploma and Advanced Diploma.

A 20-credit unit is assumed by the ICM and accordingly assessed by the Examiners, to require a total of 200 hours work.

These hours are subdivided into appropriate categories of learning opportunities, such as lectures, seminars, preparation time, directed study, time spent on assessment items and exam preparation. Approximately one quarter of this time should be devoted to formal contact time. A further one quarter of this time should relate to directed learning. The balance of workload is comprised of individual, self-managed student learning and revision.

Note that for a unit delivered in standard mode, the ICM examinations calendar accommodates a teaching year of approximately 2 x 26 teaching weeks, including 4 weeks of revision/end of year assessment. This enables teaching to take place around the 2 major exam sittings of the year being June and December and for re-sits to take place around the other exam sittings (September and March).

Furthermore, each unit is assigned to a particular 'Level', each Level corresponding to the group of subjects contained within the programmes leading to the awards of Certificate, Diploma, Advanced Diploma or Graduate Diploma.

## **5. ICM Qualifications and Progression**

---

Understanding your qualification is important and the following guidelines outline how ICM qualifications fit with the UK National Qualifications Framework and our own ICM membership programme.

ICM has benchmarked its qualifications against the UK National Qualifications Framework (NQF) and the UK government's regulator for qualifications – Ofqual (Office of the Qualifications and Examinations Regulator) ([www.ofqual.gov.uk](http://www.ofqual.gov.uk)). In recognition of the revised NQF levels and with reference to the National Occupational Standards, the Institute of Commercial Management has produced the following level descriptors which should be read in conjunction with the table provided at the end of this section in 5.5.

ICM approves a number of qualifications for entry to Associate Membership and Graduate Membership of the Institute of Commercial Management. Full details of professional membership of the ICM can be found on the website at [www.icm.ac.uk](http://www.icm.ac.uk). The appointment to Fellow of the ICM is a significant position and demands exceptional experience as well as appropriate qualifications.

### **5.1 ICM Certificates**

Entry to an ICM Certificate programme requires completion of secondary education or equivalent as specified on each course syllabus.

ICM Certificates represent a level of qualification that recognises the ability to gain, and where relevant apply a range of knowledge, skills and understanding.

ICM Certificate holders should be able to display competence in the application of knowledge in the performance of a range of work activities, some of which may be routine and predictable with some being complex or non-routine.

Learning at this level involves gaining knowledge and skills appropriate for individuals working semi-independently, or receiving basic supervision and training from others in their field of work.

Students should begin to develop a degree of individual responsibility or autonomy in their study as well as the ability to collaborate with others, for example through participation in work groups or teams.

Successful completion of the ICM Certificate enables progression in to the first year of pertinent degree programmes at a range of Universities. For a complete list of progression routes please refer to the website on [www.icm.ac.uk](http://www.icm.ac.uk).

## **5.2 ICM Diplomas**

Entry to the ICM Diploma requires completion of an ICM Certificate or equivalent programme of study.

ICM Diplomas represent a level of qualification that recognises the ability to gain, and where relevant apply a range of knowledge, skills and understanding.

ICM Diploma holders should be able to display competence in the application of knowledge in a broad range of varied work activities performed in a wide variety of contexts, most of which are complex and non-routine.

Learning at this level involves obtaining detailed knowledge and skills appropriate for people working independently, or providing basic supervision and training of others in their field of work and people wishing to go to university.

Progression is available from the ICM Diploma in to the second year of relevant degree programmes at a range of Universities. For a complete list of progression routes please refer to the website on [www.icm.ac.uk](http://www.icm.ac.uk).

## **5.3 ICM Advanced Diplomas**

Entry to an ICM Advanced Diploma requires completion of an ICM Diploma or equivalent qualification.

ICM Advanced Diplomas represent a level of qualification that involves specialist learning and detailed analysis of a high level of information, knowledge and skills in a specified area of work or study.

Students undertaking an ICM Advanced Diploma should demonstrate the ability to increase their depth of knowledge and understanding of an area of work or study to enable them to formulate solutions and responses to complex problems and situations.

Qualifications such as this are appropriate for people working as senior supervisors, professionals or managers who need to demonstrate significant levels of knowledge, a high level of work expertise in job roles and competence in managing and training others. Learning at this level is appropriate for people working in technical and professional jobs, and/or managing and developing others.

The ICM Advanced Diploma is equivalent to NQF Level 5 and these qualifications are comparable to UK intermediate Higher Education qualifications such as HNDs, Diplomas of Higher Education and Foundation Degrees.

Progression is available from the ICM Advanced Diploma in to the final year of relevant degree programmes at a range of Universities. For a complete list of progression routes please refer to the website on [www.icm.ac.uk](http://www.icm.ac.uk).

#### **5.4 Graduate Diplomas**

Students at this level study highly developed and complex levels of knowledge enabling the development of in-depth and original responses to complicated and unpredictable problems and situations.

Learning at this level involves the demonstration of high level specialist professional knowledge and is appropriate for senior professionals and managers. Indicative competencies involve the application of a range of fundamental principles across a wide and often unpredictable variety of contexts as well as the ability to perform technical or professional work activities in a variety of contexts with a substantial degree of personal responsibility and autonomy.

A qualification at this level is appropriate for people working as knowledge-based professionals or in professional management positions. This qualification is equivalent to a Bachelors degree level, and progression is available from the ICM Graduate Diploma in to postgraduate programmes such as Masters degrees. For a complete list of progression routes please refer to the website on [www.icm.ac.uk](http://www.icm.ac.uk).

#### **5.5 Relative Positioning of ICM Qualifications**

The table on the following page gives an indication of the relative positioning of ICM awards as compared with other types of qualifications and levels. Reference is made to the original UK National Qualifications Framework, though the revised levels shown in column 2 are those in currently use. In addition, the final column in the table (FHEQ) shows the “equivalent” qualifications commonly being developed and delivered in the University sector. These indicate clearly that the ICM Advanced Diploma equates to an intermediate level award (such as a Higher National Diploma) and the ICM Graduate Diploma equates to an Honours degree qualification.

National Qualifications Framework		Framework for ICM Levels	Framework for Higher Education Qualification levels (FHEQ)
Original levels	Revised levels		
<b>Level 5</b> NVQ in Construction Project Management* Diploma in Translation	<b>Level 8</b> Specialist awards	ICM professional postgraduate programmes under review	<b>D (doctoral)</b> Doctorates
	<b>Level 7</b> Diploma in Translation		<b>M (masters)</b> ,Masters degrees, postgraduate certificates and diplomas
<b>Level 4</b> NVQ in Advice and Guidance* Diploma in Management BTEC Higher National Diploma in 3D Design Certificate in Early Years Practice	<b>Level 6</b> Diploma in Management	<b>ICM Graduate Diploma</b>	<b>H (honours)</b> Bachelors degrees (BSc /BA)
	<b>Level 5</b> BTEC Higher National Diploma in 3D Design	<b>ICM Advanced Diploma</b>	<b>I (intermediate)</b> Diplomas of higher education, foundation degrees, Higher National Diplomas
	<b>Level 4</b> Certificate in Early Years Practice	<b>ICM Diploma</b>	
<b>Level 3</b> Certificate in Small Animal Care NVQ in Aeronautical Engineering A levels			<b>C (certificate)</b> Certificates of higher education
<b>Level 2</b> Diploma for Beauty Specialists NVQ in Agricultural Crop Production GCSEs Grades A*-C			
<b>Level 1</b> Certificate in Motor Vehicle Studies NVQ in Bakery GCSEs Grades D-G			
<b>Entry</b> – eg. Entry Level Certificate in Adult Literacy			

## 6. ICM Academic Calendar

### 6.1 Examinations

Examinations are externally set and marked by the Institute. Examinations are held in March, June, September and December each year. Candidates must be registered and paid-up Student Members of the Institute at the time they undertake the examinations.

Examination papers are securely distributed to ICM approved centres and scripts are graded in the UK by ICM appointed examiners.

Full details of examination dates are sent to ICM Approved Centres and students should obtain the required information from their own Centre.

## **6.2 Examination Closing Dates**

Candidates entering for examinations must ensure that their completed examination entry forms and fees reach the Institute at least eight weeks before the date of the examinations. Examination timetables indicate the closing date for receipt of entries for each sitting.

## **6.3 Examination Results**

These are distributed within twelve weeks after each examination period. Candidates are encouraged not to call the ICM direct for results as this confidential information cannot be disclosed over the telephone.

## **6.4 Subject Exemptions**

Students holding relevant recognised qualifications from other professional bodies may apply for exemptions on a subject-for-subject basis. Documentary evidence is required before any decision can be made.

## **6.5 Study Methods**

Your Teaching Centre has been provided with a detailed syllabus and reading list for each subject area. Each subject syllabus clearly defines the areas that you will be required to cover for each subject and your examination questions will be based on the areas and topics detailed for each subject. It is important to ensure that you obtain a copy of each subject syllabus from your Teaching Centre. Alternatively, this information can be obtained online at [www.icm.ac.uk](http://www.icm.ac.uk) in the "Study Centre" section.

Each subject syllabus is normally linked to one main textbook and the examiners base their questions on the contents of the nominated core text.

The subject syllabuses also give details of 'Alternative Texts' and texts recommended for further reading. It is recommended that you should read one or more of the 'Alternative Texts' in order to develop your skills and broaden your knowledge of the subject area.

## **7. Copies of past examination papers and other forms of assessment**

---

The ICM website ([www.icm.ac.uk](http://www.icm.ac.uk)) provides you with a database of recent exam papers in all subjects.

## **8. Subject advice, educational guidance and student support**

---

You are expected to be independent and to take responsibility for your own academic and personal life. However, your study centre should also provide appropriate help and assistance. Your tutors will direct your studies and ensure that you know what work you need to cover in any given unit. Seek advice from academic staff either during or after class or try to see them during their office hours.

## **9. Unit Information**

---

Covering all the major business studies areas and topics, the ICM Diploma and Advanced Diploma programmes have, over a fifteen year period, achieved international recognition and are currently undertaken by students in over 40 countries.

In addition to providing a comprehensive knowledge of each of the major business functions, the structure of the course ensures that students have the widest possible range of career, employment and higher education options open to them on successful completion of their studies.

### 9.1 Programme Objectives

To provide prospective Professional Computing students with a body of knowledge which will:

- increase their employment opportunities
- prepare them for supervisory and management careers in industry and commerce and;
- provide them with a qualification which will enable them to progress to higher level computing/IT or specialised business programmes at a later stage in their careers.

The ICM Diploma in Professional Computing & Information Processing programme is also suitable for those working in IT/Information Systems who require recognised professional qualifications for career advancement or promotion purposes.

### 9.2 Recommended Course Duration

To help calculate the duration of the programme, please refer to the section entitled 'Student Work Load' in section 4.2 of this handbook.

It is expected that a student will need at least 18 months to 2 years of full-time study at an ICM Approved Centre to complete the full Advanced Diploma programme.

### Examination Grades

Grade A - Distinction	70% and above
Grade B - Credit	60% to 69%
Grade C - Pass	50% to 59%
Grade D - Pass	40% to 49%
Grade F - Fail	39% and under

### Certification

On successful completion of all the examinations in Part 1, students are awarded the ICM Certificate in Professional Computing & Information Processing. Similarly, on successful completion of all the examinations in Parts 1 & 2 students are awarded the ICM Diploma in Professional Computing & Information Processing. Similarly, on successful completion of 12 subjects, students are awarded the ICM Advanced Diploma in Professional Computing & Information Processing.

### 9.3 Continuing Education

The Professional Computing & Information Processing programme will serve as an excellent route for students who ultimately seek full professional status and will provide first class underpinning knowledge for the mainstream UK degrees in Computing or Information Systems. Holders of an ICM Diploma in Professional Computing & Information Processing may enter the ICM Advanced Diploma programme and completion of this may enable entry to the final year of appropriate Honours degree programmes.

#### **9.4 Entry Requirements**

Students holding relevant recognised qualifications from other institutions or professional examining bodies may apply for exemptions on a subject-for-subject basis.

The Institute operates an 'Open Access' policy in respect of many of its vocational programmes but prospective students should note that the following entrance qualifications are recommended for those wishing to undertake the ICM Diploma programme:

- One GCE A-level plus Three GCSEs (or equivalents)
- A suitable BTEC National Award
- Any qualification equivalent to one or more of the above.

The minimum age for registration for this programme is 18.

Applicants over the age of 20 who do not hold one of the recommended entrance qualifications may also register for the programme providing they have been in full-time employment for a minimum of two years.

#### **Advanced Diploma in Professional Computing & Information Processing**

The minimum age for registration for this programme is 18.

- The ICM Diploma in Professional Computing & Information Processing (or an equivalent level business studies qualification)
- A suitable BTEC National Award
- Any business studies qualification equivalent to one of the above.

## **10. Course Structure**

---

### **The Certificate in Professional Computing & Information Processing**

1. General Purpose Software
2. Information Processing
3. Programming
4. Numeracy & Statistics

### **The Diploma in Professional Computing & Information Processing**

5. Number and Logic
6. Systems Analysis
7. Applications
8. Programming Project

### **The Advanced Diploma in Professional Computing & Information Processing**

9. Hardware & Operating Systems
10. Managing Computer Systems
11. Networks
12. Feasibility Study

#### **10.1 Unit Syllabuses**

Syllabuses for this programme are contained in the following sections.

**10.1.1 GENERAL PURPOSE SOFTWARE Syllabus****Unit code: GPS-0908****Level: Certificate****Credits: 20****Unit leader: PY****Pre-requisites:****Main Aims of the Unit:**

This unit introduces a range of computer packages and applications to the student. The student will be able to select an appropriate package given a specific problem and will be able to utilise off-the-shelf packages to produce solutions. Furthermore, the student is introduced to the role of operating systems in relation to such packages.

**Main Topics of Study:****A. Spreadsheet**

1. Layout of a spreadsheet screen. Help features. Loading/saving models.
2. Cell formatting. Font/size. Alignment. Column width, Row height. Boxing grouped cells. Moving/copying/deleting cell(s).
3. Data types including NUMBER, TEXT, CURRENCY, SCIENTIFIC, DATE, TIME
4. Data formatting – decimal places. Alignment. Date/time options.
5. Formulae. Cell addresses. Absolute and Relative addressing. Replicating a formula to a block of cells.
6. Functions:
  - a. Number: SUM, COUNT, AVERAGE, MAX, MIN, INT, ROUND, RANDOM, SQRT, ODD, SIN, COS, TAN, POWER.
  - b. Text: CHAR, VALUE, TRIM, RIGHT, LEFT, UPPER, LOWER.
  - c. Decision: IF, AND, OR, NOT.
 Nesting functions within functions.
7. Lookup tables and their use.
8. Design and construct spreadsheet models for a given problem. Layout design.
9. Print data, Print formulae adjusting column widths as necessary. Print with/without grid lines. Print with/without row/column labels. Fitting to a page.
10. Charts. Pie chart, histogram, bar chart, line graph. Customising output.
11. "What-if" models for the determinations of best results. Practical use in business.
12. Exporting/Importing data to/from other packages.

**B. Word Processing**

1. Layout of a word processing screen. Help. Loading/Saving documents. Multiple documents.
2. Document formatting. Margins. Alignment. Indentation. Bullet points. Page breaks.
3. Font/font size. Varying fonts/size to add effect to a document. Bold/Italics/Underline.
4. The concept of hidden embedded control characters to format. e.g. underlining.
5. Block editing. Copy/Cut blocks of text. Tabs. Changing case.
6. Search and replace in lengthy documents. Use of previous documents to create an updated version such as a newsletter, catalogue.
7. Spell check and the options IT offers. Grammar checking. Limitations of both. AutoCorrect.
8. Creating a table.
9. Mail merge. Realistic business examples of its use.
10. Quick entry of commonly use/local words/phrases using a few keys. Macros.
11. Final checks on a document. Print preview.
12. Printing. Sections or entire document. Choosing quality of print. Multiple copies.

13. Columns - newspaper style. Choosing column options.
14. Inclusion of clip art. Artistic lettering. Exporting/Importing data to/from other packages

**C. Database**

1. Concept of a database as a single system with multiple users and uses.
2. Relational databases and their advantages – single entry, consistency, integrity, redundancy, independence of data and programs.
3. Concept of normalisation of data.
4. Creation of a database. Defining the data structure. Tables/columns.
5. Key fields. Primary key. Secondary key. Foreign key.
6. Data modelling. E-R diagrams.
7. Entering data. Editing data. Searches/enquiries. SQL. Basic SQL statements - SELECT...FROM...WHERE, CREATE TABLE..., INSERT INTO...
8. Report generation. Customisation of reports. Ordering. Column totalling.
9. Printing reports.

**D. Graphics and Presentation Packages**

1. Means of producing graphics - limited drawing in word processing, charts in spreadsheets to drawing packages and commercial CAD packages.
2. Typical uses of packages by industrial/commercial users.
3. Layout of a graphics package screen. Help features.
4. Main features. Line drawing. Line selection. Boxes. Shapes. Text. Libraries of objects.
5. Manipulating the drawing features. Moving, extending, copying, rotating, hatching.
6. 3-D effects. A drawing as a series of levels.
7. Dimensioning.
8. Printing. Saving/Loading.
9. Presentation package (eg PowerPoint) as a combination of text/graphics.
10. Features of a presentation package. Creating slides and a slide show. Customising to house style.

**E. Operating Systems**

1. Purposes of an operating system. GUI operating systems. Comparison between GUI and command control.
2. General features of operating systems – folders, files. Types of files for program, data, text, email, pictures etc.
3. File manipulation – copying, moving, deleting, printing.
4. Batch files.
5. Customising the desktop.
6. Installing and customising software packages.

**Learning Outcomes for the Unit:**

At the end of this Unit, students will be able to:

1. Use a range of applications packages including spreadsheets, databases, and graphics software
2. Describe an operating system and its function
3. Apply packages to a range of real-life business problems

The numbers below show which of the above unit learning outcomes are related to particular cognitive and key skills:

Knowledge & Understanding	1, 2
Analysis	3
Synthesis/Creativity	3
Evaluation	-
Interactive & group Skills	-

Self-appraisal/Reflection on Practice -  
Planning and Management of Learning 3  
Problem Solving 2, 3  
Communication & Presentation 1, 3  
Other skills (please specify) -

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. The students are actively encouraged to form study groups to discuss course material which fosters a greater depth learning experience.

**Assessment methods weightings which enable students to demonstrate the learning outcomes of the Unit:**

3 hour examination: 100%  
Candidates will be required to answer 4 questions from the 5 given.

**Indicative Reading for this Unit:**

**Main text:**

Refer to the ICM website for learning material

**Alternative texts:**

In addition, Computer manuals for specific computer packages can be used.

**Guideline for Teaching and Learning Time (10 hours per credit)**

Lectures / Seminars / Tutorials / Workshops: 50 hours  
Tutorial support includes feedback on assignments and may vary by college according to local needs and wishes.

Directed learning: 50 hours  
Practical work/ Advance reading and preparation / Class preparation / Background reading

Self managed learning: 100 hours  
Working through practical exercises on the computer and completing assignments as required will take up the bulk of the learning time. In addition students are expected to engage with the tutor and other students and to undertake further reading using the web and/or libraries.

---

Guidelines

- Each software package should be taught at a level higher than a casual user who could pick up basic concepts simply by using it. Any person who has ever used a spreadsheet computer will know how to enter data, use simple formulae, set decimal places and total a column. For this module, candidates will need to know how to use more advanced features such as absolute addresses and their use, a wider range of formats (dates etc), look-up tables and other standard functions.
- Students should be confronted with REALISTIC business uses of the software and not just simple exercise (e.g. payroll using a spreadsheet is unrealistic except for a very small business).

- Students should also meet realistic/usable layouts that would be used in business applications. e.g. A spreadsheet layout for an annual breakdown of expenses by different departments against different types of cost would have one layout. An ongoing budget where items are listed in date order with an accumulating balance showing after each would be totally different.
- Tutors will probably obtain the best results if after students are taught the rudiments through simple exercise, they are then given business case studies partially completed and asked to finish them.

**10.1.2 INFORMATION PROCESSING Syllabus****Unit code: IP-0908****Level: Certificate****Credits: 20****Unit leader: PY****Pre-requisites:****Main Aims of the Unit:**

This unit describes the main elements in the processing of data. These include input, output and data checking. The role of the main processor (CPU) is described and different types of software are explored. Various approaches to file organisation are considered and the importance of security in the user-interface is stressed.

**Main Topics of Study:****A. Introduction**

1. Definitions of hardware, software, package, program, data, parameter.
2. Overview of the range of computers currently used from lap-top to supercomputer,
3. Configuration diagram showing how the components of a computer system are related. Difference between control signal and data flow.

**B. Input**

1. Review of methods of data capture to include OCR, OMR, MICR, barcodes, text/image scanners, magnetic stripe, voice, touch screen, keyboard, and concept keyboard.
2. Advantages and disadvantages of these methods of data capture when compared with others. In particular, a comparison of speeds, costs, user involvement, accuracy of the data received by the computer.
3. Choose the best data capture method for a given application.
4. Conditions required for each of these data capture devices to be used. A detailed explanation of the internal operation is NOT required.

**C. Data checking**

1. Difference between Validation and Verification.
2. Validation checks – range, data type, presence, sequence, ...
3. Specifying the validation possible for a GIVEN SET OF DATA.
4. Check digit systems – normally using modulo-11 and weights 6,5,4,3,2,1 etc.
  - i Specify WHICH data can/should contain check digits.
  - ii State the guarantees that check digit numbers provide.
  - iii Calculate the check digit for a given number
  - iv Validate a number which contains a check digit.
  - v The use of "X" as a check digit.

**D. Output**

1. Ways in which data can be represented – tables, lists, summary statistics, charts, textual reports.
2. Name the range of printers currently available – laser, ink jet...
3. Advantages and disadvantages of different types of printers.
4. Screen output. Its limitations.
5. Identifying whether screen or printer would be more suitable for a given application.
6. Types of plotters currently available. Comparison with printers.
7. Identifying whether a printer or plotter would be more suitable for a given application.
8. Other forms of output. COM. Microfiche. Applications where these might be used.

E. **Processor**

1. Component parts of the CPU and their functions – ALU, Control unit.
2. MAIN memory and its various forms – RAM, ROM, cache, special purpose RAM e.g. for VDU ...
3. Name registers in general use – accumulator(s), program counter (PC), Memory address (MAR), Memory data/buffer (MDR/MBR), Current instruction (CIR).
4. Fetch-execute cycle.
  - i Name the steps an instruction passes through in the cycle.
  - ii Identify how the registers are used for basic instructions. e.g. LDA Price, SUB Tax. This topic MUST be covered in detail.
5. Bus structures.

F. **Software**

1. Distinction between system software and application software.
2. Distinction between general purpose and special purpose software. Understand that the USER dictates how general purpose software is used.
3. Outline of features of operating systems including systems with multiprogramming facilities. Name different operating systems currently in use and compare in outline.
4. System software.
  - i. File management software.
  - ii Utility software. Sort file. Merge files.
  - iii Language translators. Assembler, compiler, interpreter – basic differences.

G. **Filing systems**

1. Review of current storage devices – diskette, hard disc, Winchester, flash drive/pen stick, optical devices, digital versatile disk, tape and cassette...
2. The need for buffers and their role in data transfer.
3. Definitions of storage terms – file, record, field, cylinder, track, sector, header label, inter-block gap. Distinction between storage device and storage medium.
4. The structure of data stored on a storage medium.
5. Data transfer checks. Parity and its purpose. Describe odd/even parity with specific numeric examples illustrating acceptance/failure. Cyclic redundancy check.
6. File Organisation and File Access
  - i Define organisation types – serial, sequential, indexed sequential, random.
  - ii Distinction between organisation and access. Examples of a file being accessed in more than one way.
  - iii For indexed sequential organisations – an understanding of up to 2-levels of indexing. Appreciation that not all record keys appear in the index. How ANY record can be accessed. Overflow areas.
  - iv For random access – hashing algorithms. Develop a simple algorithm for a given situation.
  - v For each of the four organisation methods, describe the PROGRAM sequence of steps to:
    - a. access a SINGLE record from the file
    - b. add a new record to the file
    - c. delete a record from the file.
7. Sequential master file update using an UNSORTED transaction file. Labelled system flow chart for this process.
8. Selecting the most appropriate file organisation for a given application.

**H. Processing**

1. Different types of processing – batch and real-time.
2. The role of batch processing with today's sophisticated systems. Examples of batch processing currently in use.
3. Real-time systems – examples of current use.

**I. Security and privacy**

1. Security defined as the safeguard of hardware, software and data.
2. Distinction between security and privacy.
3. REALISTIC methods of data security – securing against
  - i accidental damage or loss of data
  - ii deliberate sabotage.
4. Methods of achieving good privacy of data.
5. Determining the security/privacy required for a given situation.

**Learning Outcomes for the Unit:**

At the end of this Unit, students will be able to:

1. Describe the main configuration of a computer systems including input, output and processor
2. Identify different types of software including system and application software
3. Distinguish between different filing and processing systems
4. Understand the users role in data security

The numbers below show which of the above module learning outcomes are related to particular cognitive and key skills:

Knowledge & Understanding    1-4  
 Analysis                            4  
 Synthesis/Creativity        -  
 Evaluation                        3  
 Interactive & group Skills    -  
 Self-appraisal/Reflection on Practice -  
 Planning and Management of Learning -  
 Problem Solving                -  
 Communication & Presentation -  
 Other skills (please specify) -

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. The students are actively encouraged to form study groups to discuss course material which fosters a greater depth learning experience.

**Assessment methods weightings which enable students to demonstrate the learning outcomes of the Unit:**

3 hour examination:                            100%  
 (Choose any 5 questions from 8. Each question is worth 20% of the marks. )

**Indicative Reading for this Unit:**

### **Main text**

Refer to the ICM website for learning material

### **Alternative texts & Further Reading:**

Computer Science for Advanced Level by R Bradley – (Stanley Thornes)  
ISBN 0 7487 4046 5 (Fourth edition).

A Level Computing by PM Heathcote & S Langfield - (Payne Galloway)  
ISBN 1 904467 52 0 (Fifth edition)

### **Guideline for Teaching and Learning Time (10 hours per credit)**

Lectures / Seminars / Tutorials / Workshops: 50 hours

Tutorial support includes feedback on assignments and may vary by college according to local needs and wishes.

Directed learning: 50 hours

Advance reading and preparation / Class preparation / Background reading / Group study / Portfolio / Diary etc

Self managed learning: 100 hours

Working through the course text and completing assignments as required will take up the bulk of the learning time. In addition students are expected to engage with the tutor and other students and to undertake further reading using the web and/or libraries.

### **Guidelines**

---

- It is essential that candidates UNDERSTAND the elements of this syllabus. This syllabus has been detailed more than the previous version. This module is the least changed of the modules in the new scheme but there ARE differences. The following points are more warnings than guidelines based on past observations of candidates' work.
- There are many examples from past submissions where candidates clearly did not understand certain topics and were merely repeating sections from books or notes. Examples of this include drawing a configuration diagram with input device connected directly to output device and arrows showing DATA flow TO an input device.
- Questions requiring the candidate to describe the processes in a fetch-execute cycle for a PARTICULAR situation (where a memory map is given in table form) frequently produced a generalised description of the cycle with no reference to the table (what was it there for?). The result was low or no marks at all.
- Validation questions nearly always produce low marks. A particular situation is stated, often in detail, in the question and the candidate asked to DESCRIBE the validation that could be applied to THAT data. The candidate then gives a standard list from text books such as "range, format, presence..." with no reference to the data. Furthermore, a list is not a description. As above, why is the data given in the question?
- Parity is relatively easy to understand but difficult to describe. The only reliable way is to give numeric examples showing failure and passing the parity check. Questions usually ask for numerical answers but candidates do not often give them. This area clearly needs to be checked at the centre by the setting and marking of homework.
- Definitions are poorly described. e.g. Marks will not be earned for defining SEQUENTIAL if the candidate states that "data is in sequential order". These candidates often then go on to state that a SERIAL file is in "serial number order". Many candidates have not understood the concepts of

Field/Record/File because their examples have been confusing. Giving “student name AND student number” as AN example of A field cannot gain marks if the SAME phrase is used for record. Candidates frequently do not give examples, as required, from the application area stated in the question. The result is NO marks.

- When discussing a particular application of computers, tutors must encourage realistic responses. While it might be reasonable for a large institution to employ security guards and other elaborate security methods for a business, it is unrealistic to use them in a small business.
- There are a wide range of scanners yet candidates often do not specify which is used. A common mistake is in referring to magnetic or optical reading for a given device when the other is used.

**10.1.3 PROGRAMMING Syllabus****Unit code: PR-0908****Level: Certificate****Credits: 20****Unit leader: PY****Pre-requisites:****Main Aims of the Unit:**

This unit will introduce the computer programming process and enable students to design and produce programs in a high-level language of their choice. The unit will stress the importance of planning the programming approach and provides a grounding for understanding how professional programmes are developed and implemented.

**Main Topics of Study:****A. Overview**

1. The difference between different types of languages from assembler, through high level languages to 4GLs and Object-oriented languages. An outline history of the development of languages over the years.
2. Differences between Procedural and Declarative languages.
3. Summary of the stages of the translation process - Compiler and Interpreter.
4. An appreciation of assembly language code through a limited set of operation codes. How these build into a program. Writing full programs is NOT required.

**B. Program Development**

1. Overview of the stages of program development from receiving a program specification to handing over a fully tested and documented program applied to a PROFESSIONAL programmer. Candidates should have an appreciation of the difference of scale of REAL programs compared with classroom exercises.
2. Initial understanding of the problem. Checking details with the systems analyst.
3. Create algorithms for real problems.
4. Comparison of different algorithmic methods - flowcharts, pseudo-code, decision tables, Jackson charts. Advantages and disadvantages of each.
5. Define Structured programming and Construct. The three main constructs – sequence, selection, iteration. Top down programming.
6. Modular programming. How modular program can be implemented. Advantages and disadvantages of their use.
7. Prototyping. Advantages and disadvantages of its use.

**C. Input and Output of data**

1. Data types and the operations that can be performed with each.
2. Keyboard entry of data. Handling and correcting invalid data.
3. Standard output layouts. Invoices. Account ledgers.
4. Design a layout for a given situation. This must be workable in the real world.
5. Differences between output on a screen and a printer.

**D. Filing data**

1. The need for files. The consequences of filing systems NOT being available.
2. File structure. Organisation and access methods. An appreciation that a file might be accessed for more than one purpose and so determining the method of organisation.
3. Programming for serial and sequential files. Commands relating to file handling – open, close, read, write, check end of file, append.
4. Programming to locate records from the whole file that satisfy a given condition.
5. Programming to accumulate values from each record in a file.

6. Programming to merge two similar files into one.
7. Sequential Master file update from an UNSORTED Transaction file. Candidates will not be expected to program for this.

**E. Handling data in memory**

1. Variables – rules for naming variables/procedures/functions.
2. Performing calculations.
3. Decisions – IF, CASE.
4. Loops. Difference between the three main looping features and WHEN to use them.
  - i FOR/ENDFOR for fixed number of repeats.
  - ii REPEAT/UNTIL for variable repeats – the process is always executed at least once – test for exit at the end.
  - iii WHILE/ENDWHILE for variable repeats – the process may not execute even the first time in rare situations – test for exit at the beginning. Use of data TERMINATORS.
5. Arrays – 1-dimensional and 2-dimensional.
  - i Read all data into an array
  - ii Output all data from an array
  - iii Search an array for a given value
  - iv Accumulate values held in some elements of an array
  - v Use one array to find data to access another.
6. Text data – extract character(s) from a string. Joining strings.

**F. Procedures/Sub routines and Functions**

1. Use of procedures for structuring and re-use processes. Library of procedures/functions.
2. Parameter passing. Calling a procedure. How parameters are handled internally.
3. Difference between procedures and functions.
4. Standard functions such as
  - Numeric – SQRT, ABS, INT, RANDOM, SIN/COS/TAN
  - String – LEFT, RIGHT, ASC, VALUE. {The names may vary in different languages}

**G. Testing**

1. Plan the structure of test data for a program.
2. Design test data for valid and invalid situations.
3. Perform a dry run/desk check using test data and an algorithm – lay out the results in table format showing the value of variables ONLY as they change.
4. Programming to test invalid conditions AND subsequent repeated re-entry after data is rejected until it is finally accepted.
5. Testing by program. Recording the results in a test log. Levels of testing – module, program, system, user acceptance.
6. Methods of locating errors of logic – dry run, trace routines, test bed routines, temporary print commands inserted.
7. Difference between syntax and logic errors. How and when syntax and logic errors are detected and corrected.

**H. Documentation**

1. The need for documentation.
2. Documentation for the maintenance programmer.
3. Documentation for the user.
4. The reasons why user and maintenance programmer documentations are different.
5. Uses and benefits of comments/annotations in program listings.

**Learning Outcomes for the Unit:**

At the end of this Unit, students will be able to:

1. Develop, write and test high level language programs
2. Appreciate the need for low-level (Assembly) languages
3. Structure a computer program appropriately using looping, arrays and sub-routines

4. Use appropriate measures to thoroughly test programs
5. Provide effective User documentation to support programs

The numbers below show which of the above module learning outcomes are related to particular cognitive and key skills:

Knowledge & Understanding 1-5  
 Analysis 1, 3  
 Synthesis/Creativity 1  
 Evaluation 1, 5  
 Interactive & group Skills -  
 Self-appraisal/Reflection on Practice -  
 Planning and Management of Learning -  
 Problem Solving 1, 3, 4  
 Communication & Presentation -  
 Other skills (please specify) -

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. Students will spend a great deal of time involved in practical exercises on the computer to develop programs. The students are actively encouraged to form study groups to discuss course material which fosters a greater depth learning experience.

**Assessment methods weightings which enable students to demonstrate the learning outcomes of the Unit:**

- 3 hour examination: 100%  
 The paper will be split into two parts:
- Section A Compulsory section accounting for 40% of the paper. Short questions requiring high-level programming code with good syntax.
  - Section B Candidates choose 4 questions from 7, each question is worth 15% of the marks. These questions will include the design of FULL test data for a given situation, dry run of a given pseudo-code routine using given data, develop a pseudo-code algorithm for a given situation.

**Indicative Reading for this Unit:**

**Main text**

Refer to the ICM website for details of learning material available.

Appropriate manuals for the chosen programming language.

**Guideline for Teaching and Learning Time (10 hours per credit)**

Lectures / Seminars / Tutorials / Workshops: 50 hours  
 Tutorial support includes feedback on assignments and may vary by college according to local needs and wishes.

Directed learning: 50 hours

Advance reading and preparation / Class preparation / Background reading / Group study / Portfolio / Computer practical exercises

Self managed learning: 100 hours

Working through the course text and completing practical assignments as required will take up the bulk of the learning time. In addition students are expected to engage with the tutor and other students and to undertake further reading using the web and/or libraries.

---

### **Guidelines**

- This module is primarily designed to test ability in procedural programming. It is similar to the previous "Computer Programming 1" module but with a few changes including an appreciation of Assembly programming.
- A good structured language is advised such as PASCAL. BASIC may be used but candidates will be expected to initialise variables even though most versions of BASIC perform this automatically. The reason for this is to encourage good programming practice.
- Prior to the examination, candidates should have designed, written and fully tested a series of programs involving input/output, different looping features, decision, arrays, files, functions and procedures. They should have drawn up algorithms for programs with at least double loops and be aware of the difference between different types of looping facilities available in a given language. In the examination, it is expected that language syntax will be correct.
- Candidates should be able to design test data to test a program fully. Dry running/desk checking a program with test data should be practised using a systematic tabular format. An appreciation that dry-running is simulating the actual running of the program by the computer.
- A clear understanding of different types of documentation is required and why different documentation is needed by different people and in different situations particularly in large commercial applications where the documentation may be extensive. Some previous work submitted for printout design has been poor because it was totally unrealistic – did not allow for multiple pages, no headings... Similarly, form designs have omitted obvious features such as the person/address to which the form should be sent.
- Tutors need to "tick off" elements of this syllabus covered as they teach it (which may not necessarily be in the order given below) to avoid missing parts of it.
- The examination will include a compulsory section of 40% for which the answers will require high-level programming code. These questions will be short routines. A candidate who has successfully written programs involving input/output, decisions, loops, arrays, text manipulation and file handling should be able to score close to full marks in this section. In the past, many candidates have rushed into the questions and produced very long routines, some of which was often irrelevant and this would have been obvious had they read the question carefully. It is possible to solve the problems with SHORT routines. Often the question states that the data is ALREADY in variables or arrays and so coding is not required for its input but this is often ignored.
- Students must be encouraged to think HOW they are going to solve a particular problem before actually doing it. Many candidates have scored under 10 out of 40% in the past on the compulsory programming section. Similarly, algorithms have sometimes been ill-thought out. A sample problem is given at the end of this syllabus to show the level of programming required.
- Dry runs should be easy marks for a systematic candidate. Few candidates in past examinations have actually predicted the correct outcomes using a given algorithm and test data.

### **EXAMPLE OF A LOGIC PROBLEM AND THE EXPECTED ANSWER**

#### **QUESTION**

**A library allows borrowers to keep books for 14 days. Input a borrowing date (as D, M, Y) and output the date the books are due back (14 days later) taking into account leap years.**

Integer array MONTH has cells 1 to 12 and holds data that shows the number of days in each of the 12 months of the year for a NON-leap year.

---

Position	1	2	3	4	5	6	7	8	9	10	11	12
Value	31	28	31	30	31	30	31	31	30	31	30	31

Assume the data is ALREADY in the array.

**NOTE** – questions often state the data is ALREADY in the array but in the past many candidates have ignored this and written routines, often incorrect, which earned NO marks and wasted valuable exam time. On occasions answers have consisted of ONLY this unneeded coding. Candidates are also asked NOT to change the variable names given in the question but they frequently do, sometimes leading to confusion when they use the variable name given in the question for a different purpose.

---

**SOLUTION**

**A pseudo-code algorithmic solution is given here – HIGH LEVEL CODING OF THIS WOULD BE NEEDED IN THE EXAMINATION. Clear indentation is used here to show start and end of multiple-instruction routines such as IF..ENDIF and should be used for loops as well.**

**INPUT (D,M,Y)**

**IF Y divisible by 4 ( \*\*\* see below ) testing for leap year**

**Then MONTH[2] ← 29 to allow for leap year or not, reset February**

```

    Else MONTH[2] ← 28
ENDIF
D ← D + 14
IF D > MONTH [M]
    Then D ← D – MONTH[M]
        M ← M + 1
year?
        IF M > 12
            Then M ← 1
                Y ← Y + 1
        ENDIF
ENDIF
OUTPUT (D,M,Y)

```

14 days later, but this might be into a new month?  
so check against the MONTH table.  
and if so, move day into next month  
and move month on by one – but it could be new  
if so, move month to January  
and move year on to the next year.  
corrected now date output.

**\*\*\* Might be programmed as IF INT(Y/4)\*4 = Y or IF Y MOD 4 = 0**

Note that this routine is relatively short and would be only slightly longer in a high level language. Approaching this problem, the candidate should note different situations that might occur. In particular is 14 days later

- still in the same month?
- into the next month but still same year?
- into the next month which is then into the next year?

The candidate should recognise at first reading :

- there are no loops
- the month table is already in memory (perhaps by some previously written routine) and the setting up of its initial values is NOT needed.
- decisions (IF statements) WILL be needed to determine
  - a) whether the year is a leap year and what to do about the table,
  - b) how to test for three different date types listed above.

These points will take a few minutes but if thought out in advance should mean that the solution could be written out immediately. The only problem might be that some candidates would not know how to check for divisibility as suggested above. In such situations, it is acceptable for the candidate to write a pseudo-code equivalent to complete the logic AND make a note "I do not know how to do this". Such honesty would be refreshing and would lose only one mark of the whole. Unfortunately, some candidate's grasp of syntax has been so poor that the whole solution has been written in note form like this with statements like "add total to the array". This clearly shows that candidates had little idea how to program for arrays.

**10.1.4 Numeracy and Statistics Syllabus**

**Course Title** Numeracy and Statistics

**Unit Code** NS-0605

**Level** 1

**Credits** 20

**Unit Leader** GW

**Pre-requisites**

**Main Aim(s) of the Unit:**

The aims of this unit are to equip students the necessary skills to interpret information, carry out calculation and interpret the results of that information and later, to equip students with some of the analytical skills that are relevant to the study of Business Studies as a whole.

**Main Topics of Study:**

<b>Using a Calculator:</b>	Approximations / Estimation / Degrees of Accuracy / The Keys of a Calculator / Standard Form
<b>Fractions:</b>	Types of Fraction / Equivalent Fractions / Operations Involving Fractions / The Conversion Between Fractions & Decimal Fractions /
<b>Ratio and Proportion:</b>	Ratio / Division in a Given Ratio / Direct Proportion / Scale Diagrams & Models
<b>Measurement:</b>	Metric and Imperial Units / Conversion Between Metric & Imperial Units
<b>Percentages:</b>	Percentages / Finding a Percentage of an Amount / Increasing & Decreasing an Amount by a Given Percentage / Expressing One Quantity as a Percentage of Another
<b>Wages &amp; Salaries:</b>	Basic Pay / Overtime Rates / Commission / Piecework / Deductions from Pay
<b>Travel:</b>	Foreign Currency / Time / Timetables
<b>Sampling, Surveys, Questionnaires:</b>	Surveys / Censuses / Samples / Sampling Methods / Bias / Questionnaires / Pilot Surveys / Hypothesis Testing
<b>Clarification &amp; Tabulation of Data:</b>	Tabulation / Classification of Data / Tally Charts / Frequency Tables
<b>Statistics on Display:</b>	Pictorial Representation of Data / Pictograms / Bar Charts / Pie Charts / Line Graphs / Frequency Polygons / Drawing Inferences from Diagrams
<b>Averages &amp; Range:</b>	The Arithmetic Mean / The Mode / The Median / The Use of Mean, Mode & Median / The Mean & Median of a Frequency Distribution / Range
<b>Cumulative Frequency:</b>	The Cumulative Frequency Curve (or Ogive) / The Median / The Interquartile Range / Percentiles
<b>Probability:</b>	Introduction / Probability From Theory and Experiment / Simple Probabilities / Simple Laws of Addition / Possibility Space
<b>Networks &amp; Decision Trees:</b>	Networks / Decision Trees
<b>Algebra:</b>	The Basics of Algebra / Indices / Brackets / Common Factors / Equations / Harder Equations / Trial & Improvement / Flow Diagrams / Sequences / Formulae
<b>Graphs:</b>	Graphs & Curves / The Interpretation of Graphs / Graph Plotting / Conversion Graphs / Cartesian Coordinates / Straight Line Graphs / Gradients & Intercepts

<b>Geometrical Shapes:</b>	Lines & Angles / Polygons / Circles & Angles / Tessellations / 2D Representation of 3D Objects
<b>Mensuration:</b>	Perimeters of Polygons / Area / The Circumference and Area of a Circle / Volume
<b>Time &amp; Travel Graphs:</b>	Distance-Time Graphs with Straight Lines / Distance-Time Graphs with Curves / Velocity-Time Graphs / Growth & Decay
<b>Further Algebra:</b>	Simultaneous Equations / Inequalities / Removing Brackets / Rearranging Formulae

**Learning Outcomes for the Unit**

At the end of this Unit, students will be able to:

- 1 Identify appropriate types of quantitative information and their sources.
- 2 Rationalise and structure that information.
- 3 Use and analyse information that is presented in numerical form in an appropriate method to extract all relevant trends, patterns and points of importance.
- 4 Present the results and data informative and attractive manners including in diagrammatic form to clients, partners and business colleagues.

**The numbers in the boxes below show which of the above unit learning outcomes are related to particular cognitive and key skills.**

Knowledge & Understanding	Analysis	Synthesis /Creativity	Evaluation	Interactive & Group Skills	Self-appraisal/ Reflection on Practice
1-3	3	3,4	1,2,3	-	-
Planning and Management of Learning	Problem Solving	Communication & Presentation	Practical/Psycho-motor Skills	Other skills (please specify):	
-	1-4	4	4	-	-

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. The students are actively encouraged to form study groups to discuss course material which fosters a greater depth learning experience.

**Assessment methods which enable student to demonstrate the learning outcomes for the Unit:**

**Weighting:**

3 Hour Examination

100%

**Indicative Reading for this Unit:**

**Main Text:**

Application of Number - B Gaulter & L Buchanan (Oxford University Press)

**Guideline for Teaching and Learning Time (10 hrs per credit)**

50 hours **Lectures / Seminars / Tutorials / Workshops**

Tutorial support includes feedback on assignments and may vary by college according to local needs and wishes.

50 hours **Directed learning**

Advance reading and preparation / Class preparation / Background reading / Group study /

Portfolio / Diary etc

100 hours **Self managed learning**

Working through the course text and completing assignments as required will take up the bulk of the learning time. In addition students are expected to engage with the tutor and other students and to undertake further reading using the web and/or libraries.

**10.1.5 NUMBER AND LOGIC Syllabus****Unit code: NL-0908****Level: Diploma****Credits: 20****Unit leader: PY****Pre-requisites:****Main Aims of the Unit:**

This unit provides an overview of the numeric processes that computers perform and also some of the numerical methods that are commonly solved by computers. The conversion of values from one number system to another will be introduced and students will be able to perform arithmetic operations totally in both binary and in hexadecimal. Other approaches to handling numbers are also considered including iterative methods, number progressions and basic statistical methods.

**Main Topics of Study:****A. Number systems**

1. Definition of a general number system. The common rules of decimal, binary, octal and hexadecimal. From these, formulate the general rules for a number system with base N.
2. Definition of bit, byte, character, word as basic storage units.
3. Methods to convert between decimal, binary, octal and hexadecimal integers.
4. Methods to convert between decimal, binary, octal and hexadecimal fractions.
5. The special relationship between binary and octal and hexadecimal.

**B. Arithmetic operations**

1. Addition, subtraction, multiplication and division performed TOTALLY within with binary, octal or hexadecimal systems.
2. Given that 10 is approximately 2 to the power 3 1/3, work out how many binary places are equivalent to a given number of decimal places.

**C. Use of memory**

1. Representing binary numbers in bytes and words. Normally only 8 or 16 bit-words will be used in numeric questions to reduce the working.
2. 2's complement method of holding negative numbers. The ACTUAL NUMERIC VALUE of the sign bit in a 2's complement number.
3. Representing fractional values in memory. Floating point numbers. Normalisation of floating point numbers. Determine how a decimal number would be held in binary in float point form including negative numbers. Determine the decimal value of a floating-point binary number.
4. Fixed point representation of mixed integer and fractional numbers – normally the mid-point will be the implied binary point.
5. How memory holds non-numeric data. ASCII. Binary Coded Decimal. Methods of holding variable length string data.
6. Format of an instruction in a binary word. A simplified form is normally tested which includes (i) operation code, (ii) register number and (iii) single address.

**D. Matrix notation**

1. How a 2-D matrix is held in 1-D memory.
2. Basic rules for adding, subtraction and multiplying matrices.
3. Develop an algorithm for adding, subtracting or multiplying two matrices.
4. Determine the inverse of a 2x2 or 3x3 matrix.
5. Matrix method of solving simultaneous equations.

**E. Iterative methods**

1. Iteration as the idea of “homing-in” to provide an accurate answer as required. Understand how far to go to determine an answer to the required number of figures/decimal places.
2. Understand that an iterative equation could converge to an answer or diverge away from it. Means of determining whether a particular iterative equation will converge for a given problem. e.g. An equation may have two solutions (near  $x=2$  and  $x=4$ ). To solve for the solution near  $x=2$ , one particular iterative equation might converge near  $x=2$  but another might diverge or home in on the solution near  $x=4$ .
3. Practical applications of iteration such as:
  - i Newton-Raphson method to determine the square root of a number.
  - ii Determine the reciprocal of a number ( $1/N$ ).
  - iii Solve an equation up to degree four by an iterative method.
  - iv Solve simultaneous equations by an iterative method.

**F. Other Numerical methods**

1. Graphical method of finding the “best fit” (linear programming). Determine inequalities in a linear programming problem. Plot suitable lines graphically to represent the built-in restrictions. Plot a suitable line to maximise or minimise (e.g. minimum costs or maximum profit). Alternatively, candidates can use the Simplex method to solve a problem.
2. Venn diagrams. Application to real problems.
3. Apply a given process to determine the best fit. The method will be defined on the examination papers. e.g. best route to take between different points.

**G. Financial**

1. Arithmetic and geometric series. Determine the  $n^{\text{th}}$  term and sum of  $n$  terms for each.
2. Application to financial situations – discount and depreciation. Inflation.
3. Interest – simple and compound.
4. Economic order quantity (EOQ).

**H. Statistics**

1. Averages – definitions of Mean, Median, Mode, Range, Inter-quartile range, frequency.
2. Calculation of averages for a set of numbers including data held in a frequency table. Determine the best average to use in a given situation.
3. Calculation of averages for tabular data with class intervals.
4. Dispersion. Standard deviation. Skewness. Normal distribution. Correlation treated non-numerically.
5. Probability. Definition. Simple problems involving:
  - i Mutually exclusive events – the probability of either occurring.
  - ii Independent events. Probability of both occurring.
  - iii Conditional probability– probability of  $x$  given  $y$  has occurred =  $P(x|y)$ . Bayes theorem.
6. Expected value.
7. Permutations and combinations. Simple problems.

**Learning Outcomes for the Unit:**

At the end of this Unit, students will be able to:

1. Handle binary and other number systems related to computer usage
2. Describe how data is held in memory
3. Use iterative approaches to solve simple mathematical problems
4. Solve problems using techniques such as linear programming and Venn diagrams
5. Use basic statistics in the description and analysis of data

The numbers below show which of the above module learning outcomes are related to particular cognitive and key skills:

Knowledge & Understanding 1-5  
 Analysis 3, 4  
 Synthesis/Creativity -  
 Evaluation 4

Interactive & group Skills -  
Self-appraisal/Reflection on Practice -  
Planning and Management of Learning -  
Problem Solving 3, 4, 5  
Communication & Presentation 5  
Other skills (please specify) -

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. The students are actively encouraged to form study groups to discuss course material which fosters a greater depth learning experience.

**Assessment methods weightings which enable students to demonstrate the learning outcomes of the Unit:**

3 hour examination: 100%  
*(Answer 5 questions from 8, each question worth 20% of the marks)*

**Indicative Reading for this Unit:**

**Main text**

Refer to the ICM website for learning materials

**Alternative Texts & Further Reading:**

Quantitative Techniques by T.Lucey (Thomson)  
ISBN 1844801063 (Sixth Edition)

Computer Science for Advanced Level by R Bradley (Stanley Thornes)  
ISBN 0 7487 4046 5 (Fourth edition).

**Guideline for Teaching and Learning Time (10 hours per credit)**

Lectures / Seminars / Tutorials / Workshops: 50 hours  
Tutorial support includes feedback on assignments and may vary by college according to local needs and wishes.

Directed learning: 50 hours  
Advance reading and preparation / Class preparation / Background reading / Group study / Portfolio / Diary etc

Self managed learning: 100 hours  
Working through the course text and completing assignments as required will take up the bulk of the learning time. In addition students are expected to engage with the tutor and other students and to undertake further reading using the web and/or libraries.

### Guidelines

- This syllabus is designed to provide a test of the numeric processes that computers perform and also test some of the numerical methods that are commonly solved by computers. It has been extensively revised from the former Computer Arithmetic and Logic module.
- Note that Boolean algebra has been moved to the Hardware and Operating System module to include technical aspects of logic circuits.
- Candidates should have had much practice in converting between number systems using a variety of different methods. In particular it is important that a FAST EFFICIENT METHOD can be chosen.
- Candidates must be able to perform arithmetic operations totally in both binary and in hexadecimal. Too many answers in the past have converted binary to decimal, performed decimal arithmetic and then converted back. Apart from the extra time this takes in the examination, the additional work inevitably produces more mistakes.
- Much time will be saved if candidates actually know, for instance, that the sign bit in a twos-complement has a particular negative value. Also, multiplying and dividing by powers of 2 is effectively the same process as in decimal where additional noughts or movement of the decimal point is all that is needed for powers of ten. Many past answers, normally wrong, have taken a page of calculations to divide a binary number by decimal 4 when in fact the answer can be written down in a few seconds by shifting the digits to places.
- Candidates must ensure that when questions particularly relate to numbers held in memory, (the word size will be given) that the answer shows ALL the bits including leading zeroes as it would be held in that memory. This is particularly relevant where negative numbers are being held in twos-complement format.
- Candidates are only expected to be able to solve simple problems in statistics. Tutors should explain the meanings of the results of a statistical calculation.

**10.1.6 SYSTEMS ANALYSIS Syllabus****Unit code: SA-0908****Level: Diploma****Credits: 20****Unit leader: PY****Pre-requisites:****Main Aims of the Unit:**

This unit describes the processes involved in the analysis, design and implementation of new or upgraded computer systems. The unit will explore the basic stages involved in the systems life-cycle. Other issues will be considered including privacy & security, the importance of appropriate and comprehensive documentation, and characteristics of common data handling systems.

**Main Topics of Study:****A. Overview**

1. The need for systems analysis for business systems.
2. The role of the systems analyst.
3. The role of other people in defining a new system and updating an existing one.
4. Differences between the development of a new system and a MAJOR upgrade of an existing system.
5. SSADM – an overview. CASE tools.

**B. Systems Life Cycle**

1. The sequence of stages in the life cycle. Time scales.
2. Initial Investigation.
3. Feasibility Study. Initial briefing statement. Contents of the report. Possible management decisions on reading and discussing the report. Cost/Benefit analysis. User needs.
4. Analysis stage. Investigations into current systems through questionnaires, interviews, observation of everyday usage, documentation. Evaluation of the relative merits of these methods. Who to interview. Techniques for maximising information. Identifying the main processes needed. Recording the results.
5. Design stage.
6. Implementation stage. Program production or acquisition. Testing methods. Training. Documentation completion. Hand-over. For system upgrade - methods of file conversion and changeover.
7. Maintenance & Review. Reasons for each. Differences between them. How each is completed and by whom and when.

**C. Recording the findings for the system.** Students may be required to draw one of the charts/tables for a given system during the examination.

1. E-R diagrams and associated terms.
2. Data Flow diagrams.
3. System flow-charts.
4. Decision tables.
5. Progress/Testing charts. Gantt charts.
6. Deciding which is the most appropriate chart to use.

**D. Design**

1. Deciding the most appropriate data capture method for a given system.
2. Data-entry form design. Factors that makes a form effective and user friendly?

3. The need for data to be coded. The advantages of data coding. Types of data codes – sequence, block, faceted, hierarchical. Coding data for a given application. Check digit systems.
4. Output design. Making screen outputs effective and user friendly. Limitations of screen output.
5. Printout design. Standard outputs (e.g. the invoice, account statement).
6. Identify necessary files, organisation method and access method. Identify files that must be accessed for different purposes and the subsequent effect this has on organisation method.
7. Identify contents of each file.
8. Plan and schedule the development process. Deciding the method of obtaining programs and language/platform to be used.

**E. Implementation**

1. Analyst initial briefing of the programming team. Subsequent collaboration between analyst and programmer. Time scales and planning.
2. Analyst role in testing. Test plan. Design of test data. Testing log. Testing at different levels – module, program, system, and user-acceptance.
3. Training of users. Production of documentation.

**F. Software acquisition methods**

1. Purchase and customise standard software.
2. Develop a database and associated processes.
3. Commission a software house.
4. Write in-house programs.
5. Prototyping.
6. Comparison of the above methods – advantages and disadvantages.

**G. Security and Privacy**

1. Define Privacy as the need to build in safeguards for the data subjects.
2. Define Security as the need to safeguard hardware, software and data.
3. Building REALISTIC security into a system. The differences between small and large business security systems.
4. Use of passwords and encryption for privacy.
5. Methods to secure hardware, software and data.

**H. Documentation**

1. System specification and its contents.
2. Program specification and its contents.
3. User documentation and its contents. Maintenance documentation and its contents. The differences between these three.

**I. Characteristics of common systems.**

1. Describe the main elements of the following processes:
  - i file sort,
  - ii merging of files,
  - iii search a files for records which meet a given condition,
  - iv produce a printed report following a search.
2. Identify method of data capture, main processing, required outputs, files (contents, organisation and access), manual processes and people involved for:
  - i Stock control
  - ii Retail systems
  - iii Mail-ordering systems
  - iv Staff payroll and personnel records
  - v Medical records
  - vi Library administration
  - vii Club membership.

**Learning Outcomes for the Unit:**

At the end of this Unit, students will be able to:

1. Describe the stages of computer systems development including analysis, design, development and implementation.
2. Evaluate appropriate approaches to software acquisition.
3. Compare appropriate privacy and security measures in different systems.
4. Describe the main elements of file processing across a range of applications.

The numbers below show which of the above module learning outcomes are related to particular cognitive and key skills:

Knowledge & Understanding 1, 2, 4  
 Analysis 2, 3  
 Synthesis/Creativity 4  
 Evaluation 2  
 Interactive & group Skills -  
 Self-appraisal/Reflection on Practice -  
 Planning and Management of Learning -  
 Problem Solving 2, 3  
 Communication & Presentation -  
 Other skills (please specify) -

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. The students are actively encouraged to form study groups to discuss course material which fosters a greater depth learning experience.

**Assessment methods weightings which enable students to demonstrate the learning outcomes of the Unit:**

3 hour examination: 100%  
 (Answer any 5 questions from 8, each worth 20% of the marks)

**Indicative Reading for this Unit:**

**Main text**

Refer to the ICM website for notes on this subject

**Alternative Texts & Further Reading:**

Systems Analysis and Design by Donald Yeates and Tony Wakefield (Prentice Hall) – 2nd edition ISBN 0273 65536 1. 499 pages.

Computer Science for Advanced Level by Ray Bradley – (Nelson Thornes) 4<sup>th</sup> edition ISBN 0 7487 4046 5

**Guideline for Teaching and Learning Time (10 hours per credit)**

Lectures / Seminars / Tutorials / Workshops: 50 hours

Tutorial support includes feedback on assignments and may vary by college according to local needs and wishes.

Directed learning: 50 hours

Advance reading and preparation / Class preparation / Background reading / Group study / Portfolio / Diary etc

Self managed learning: 100 hours

Working through the course text and completing assignments as required will take up the bulk of the learning time. In addition students are expected to engage with the tutor and other students and to undertake further reading using the web and/or libraries.

---

### **Guidelines**

---

- Tutors should endeavour to use case studies to teach as much of this syllabus as possible.
- Candidates will be expected to have some appreciation of REAL business systems rather than be able to repeat generalised notes gained from books. For instance, a consideration of time scales should make it clear that the major cost and delays in a system occur in the production of programs however they are produced.
- A clear understanding of file organisation and access methods needs to be developed by the candidate.
- Many elements of this syllabus can be helped by setting exercises for each student or for a group of students.

**10.1.7 APPLICATIONS Syllabus****Unit code: AP- 0908****Level: Diploma****Credits: 20****Unit leader: PY****Pre-requisites:****Main Aims of the Unit:**

This unit will introduce a wide range of business applications and will improve the students understanding of such systems in a real situation. Applications of systems involving a large number of records will be introduced. The student will gain a good understanding of the meaning of data in different contexts. *(Note: This unit does **not** examine in detail the use of off-the-shelf applications packages such as spreadsheets, databases, etc.)*

**Main Topics of Study:****A. General**

1. Define package, program, application, program and data.
2. Benefits and limitations of different data capture methods to the business and to the client.
3. Data capture forms (screen and paper). In this case, an awareness that screen is an output device confirming inputs.
4. Outputs (printed and screen). Selection of appropriate form of output. Billing layouts.
5. Distinction between on-line direct access and batch runs.
6. Process diagrams. Systems flowchart or Data flow diagrams could be used but a formal diagram is not expected – the requirements are that candidates are aware of the sequence of events including rectifying errors.
7. Back-up systems needed including restart processes where a lengthy print run may fail in mid run because of a printer failure.

**B. Applications:** Thorough investigation of each of the following applications:

- a. Mail order and customer accounts
- b. Stock control
- c. Supermarket sales
- d. Library administration
- e. Public utility companies' administration
- f. Hotel administration and bookings
- g. Banking related to customer accounts.
- h. Club membership
- i. Estate agency (real estate) administration.

Such an investigation of each of the above applications should cover the aspects listed below.

1. Overall purpose of the application.
2. The role of the people involved.
3. The detailed format of the sources data for each aspect of the application.
4. The method of data capture for each type of input. Devices suitable for this.
5. The processes that operate in the business – systems diagrams. Batch or on-line.
6. The filing systems required including contents of each file and methods of organisation and access.
7. The outputs that are produced.
8. The devices required for each type of output.
9. How each output will be used and possible follow-ups.

**Learning Outcomes for the Unit:**

At the end of this Unit, students will be able to:

1. Evaluate a specific business system
2. Identify required inputs and outputs
3. Analyse the data format requirements of systems
4. Describe the required processing for a complete business system
5. Identify the roles of people involved in the system

The numbers below show which of the above module learning outcomes are related to particular cognitive and key skills:

Knowledge & Understanding	2, 4
Analysis	3
Synthesis/Creativity	4
Evaluation	1, 3
Interactive & group Skills	-
Self-appraisal/Reflection on Practice	-
Planning and Management of Learning	-
Problem Solving	3, 4
Communication & Presentation	4
Other skills (please specify)	-

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. The students are actively encouraged to form study groups to discuss course material which fosters a greater depth learning experience.

**Assessment methods weightings which enable students to demonstrate the learning outcomes of the Unit:**

3 hour examination: 100%  
 The examination paper will consist of 8 questions with question 1 being compulsory and a further four being required.

**Indicative Reading for this Unit:**

**Main text**

Refer to the ICM website for notes on this subject

**Alternative texts and Further Reading:**

Computer Science for Advanced Level by R Bradley – Stanley Thornes

**Guideline for Teaching and Learning Time (10 hours per credit)**

Lectures / Seminars / Tutorials / Workshops: 50 hours

Tutorial support includes feedback on assignments and may vary by college according to local needs and wishes.

Directed learning: 50 hours

Advance reading and preparation / Class preparation / Background reading / Group study / Portfolio / Diary etc

Self managed learning: 100 hours

Working through the course text and completing assignments as required will take up the bulk of the learning time. In addition students are expected to engage with the tutor and other students and to undertake further reading using the web and/or libraries.

---

## Guidelines

---

- **WARNING.** This module is mainly concerned with systems of particular business situations. Candidates should only be entered for this module if a range of applications are studied and in detail. A summary in note form of a sample system is listed at the end of this syllabus to give an outline of the requirements.
- Candidates must be introduced to the idea that business applications normally involve a very large number of records (customers, accounts, products) and must not be confused with classroom exercises which tend to consider only a small number. A supermarket could have 25,000 products on its shelves. An electricity board may supply power to one million homes and businesses. There may also be a large number of different processes within an application, many of them as fail-safe provisions to correct for errors or change of circumstances. "Display the file on the screen" is clearly not even a remote possibility, even with scrolling.
- Candidates will be expected to have a good understanding of the particular data associated with each application area. In addition, there may be a variety of different types of inputs and outputs. Stored data may also be used in different ways. Answers to questions must be specific. For instance, they must go beyond stating "the data is input..." by indicating exactly what data is captured, how, when and by whom/ what.
- Candidates who rely on "learning" the processes will find this module difficult – they must understand the whole system. The supplied example below (only in note form) shows the extent to which the real system differs from the ideal one where everything works without fault. The applications considered are complex. In a given paper, different aspects of business data processing will be covered in different applications.
- Online applications which use the internet will NOT be tested on this paper but will be tested in the NETWORKS paper.
- Each application should be taught separately, preferably as a case study. Teachers may find that role-play may be an effective way to teach this subject with students acting as customers and data clerks and even being a particular file. The movement of data can then be illustrated.
- The systems considered here are not part of an academic exercise. Many will be relevant to their home life and therefore a better understanding can be gained by discussing this outside the classroom with affected people – parents, family and friends.
- It is not sufficient to cover a small number of the applications because this will restrict the candidate's choice in the examination and the chosen ones may not be tested in a given paper.
- Five to six hours on each application should be sufficient. It may take ten hours to cover the first. Methods learnt in one application will almost certainly help understanding of later ones which should therefore take less time. Well into the course, one of the systems could be set as a student homework project provided the teacher later brings out the elements of the solutions for all the class to see.

**EXAMPLE (IN OUTLINE) – Public Utility – ELECTRICITY SUPPLIER (ADMINISTRATION)**

**Purpose :** To accept new customers, correct/amend customer details, update customer records. Read customer meters , bill them and accept payment. Generate reports for management. Maintain past records for enquiry purposes.

**People:** Office staff will enrol new customers and deal with queries. Meter-readers will read meters in houses. Computer staff will update files from these readings and subsequently produce bills that will be despatched to customers. Customers will pay (or not). Office staff will deal with payments and request non-payment runs that will generate reminder letters.

**Source data:**

- New customers – Name/Address/Telephone/Start date/Existing meter No and reading if one already present in the house. Some of this data may not be available at the time of registration.
- Change of customer details, Enquiry OR Customer cancelling – letter or telephone call with details of the account to verify authenticity.
- Meter reading – electricity usage in the household from meter reading and meter number.

**Data capture:**

- New customers/Changes – Keyboard using on-screen form.
- Meter reading:
  - a. Palm top – new reading keyed in. Checked against expected reading/past usage.
  - b. Mark Sense form – mark sense reader with batches of readings per meter-reader.
  - c. Manual forms – keyboard entry at the office.
  - d. Customer reading – some companies accept meter readings by telephone/internet made by the customer but require an official reading perhaps once per year.

**Processes :**

- New customer entry/enquiry/changes to existing customer – update the customer file. This includes customer readings and reported errors. Generate new bill for a leaving customer.
- Create from file, meter-round details for the meter-reader – file down-load onto palm-top, mark documents or printed forms – each in round order.
- Input of meter readings to update accounts for billing.
- Bill production by batch processing - exceptions on request
- Payment recording – either direct to account from OCR form/keyed payment or batch of payment records from remote office.
- Batch non-payment check.

**Filing systems:**

- Customer file
  - identity details: name/address lines/telephone/start date/payment method/round number/number within the round/account number
  - meter details: meter number/previous reading/particular tariffs
  - history : past usage details.
  - Access: sequential for batch billing run  
direct access for individual enquiries, payments, meter reading update.
  - Organisation : Index sequential to achieve both access methods.
- Meter readings file: (to update customer file)

Customer address/account number/round number/meter reader ID/expected reading (prediction – to reduce errors when reading) /actual reading (filled in on the round)

Organisation & Access: Sequential (number within round) for recording new readings.

- History file (customer past usage)

**Outputs & Devices:**

- Billing invoices with payment slip – printed
- Customer records for updating/enquiry – screen
- Reminder letters for customers & lists for accounts department – printed.

**Use of Outputs:**

- Screen records for clerks to deal with problems and answer customer queries.
- Bills for customer to identify reasons for charging and make payments. Data clerks to accept payment.
- Reminder letters for customer to take remedial action and use for late payments.

This is only an outline. The candidates should be taken through the whole process, preferably in a logical sequence from first registration of a customer to payment of bills. The scale of the problem should be discussed to show one customer is only one of a large number. The following and many more questions should be discussed and answered. There are no precise answers but those that are put forward should be realistic within the confines of this application and situation.

- How can the company cope with say, 1 million customers?
- How often are bills printed? If this is three monthly (50 working days allowing for other processes and building in safety margins), all of the following considerations need to take this into account.
- How many meter readers might be needed assuming that the reading is staggered over the three months? It therefore follows each would need their own data capture device and appropriate file down-load for each round, prepared in advance.
- How long does it take, realistically, for a meter-reader to move from one house to the next and read a meter? Simulate this process to find out.
- How do you deal with house which the meter-reader cannot gain access on his round? What happens to subsequent billing? Candidates may have experience of this or be able to talk to parents about it.
- How are businesses treated differently from homeowners? Business may have the equivalent of 10, 100 or 1000 times the electricity usage and have multiple meters.
- How many operators (and therefore computer terminals) are needed to deal with queries and changes?
- What hardware would be needed to print 1 million bills over three months, again staggering them over three months?
- How do you post 1 million bills over the three months? The letters would need to be addressed. If the customers are all local (large city) it may pay to employ delivery staff otherwise general mailing systems would be used.
- What security is needed to ensure continuity? Even temporary failure of a fast main printer could have major delaying effects. How can this be overcome? Failure midway through a run means some of the bills are printed and the file updated while the remaining customers are effectively unprocessed. How can the system cope with this problem?
- Past data is filed in a history file. How far back in time should this go? How frequently should the history files be updated?
- How are reminder letters generated? If just 1% fail to pay on time, this is 10,000 over three months. This would be 200 per day over 50 days. How does the system identify this group? Should the reminder run be performed once per week/once per month - this would delay income? Clearly they would have to be batched and related to when the meter was read/bill sent.

The questions are almost endless. When any of these systems are initially created, such questions will face the system designers. This is therefore a good way to teach this module encouraging enquiring minds to look for solutions (and problems). Students will have a better understanding and this will almost certainly simplify the consideration of other systems.

**10.1.8 PROGRAMMING PROJECT Syllabus**

**Unit code: PP-0908**

**Level: Diploma**

**Credits: 20**

**Unit leader: PY**

**Pre-requisites:**

**Main Aims of the Unit:**

The aim of the project is to facilitate the application of programming skills in the development of a real-life situation (normally related to local business) and to extend the student's experience of programming.

The student will identify a specific complex problem and produce a working solution to that problem.

**Main Topics of Study:**

Students will build on the knowledge gained in earlier units and this project will enable the demonstration of a range of knowledge and skills including:

- project identification
- Analysing a real-life situation
- Designing the system in relation to specific elements such as input, output, file structures and procedures
- Developing a computer program in a suitable programming language
- Incorporating test procedures for the program
- Producing appropriate documentation

**Learning Outcomes for the Unit:**

At the end of this Unit, students will be able to:

1. Analyse a given business problem or issue
2. Design an appropriate system to resolve the identified problem(s)
3. Develop a program using an appropriate high-level language
4. Design and develop appropriate documentation for the system
5. Evaluate the complete process and provide recommendations for further enhancements

The numbers below show which of the above module learning outcomes are related to particular cognitive and key skills:

Knowledge & Understanding	2, 3, 4
Analysis	1, 5
Synthesis/Creativity	1, 5
Evaluation	5
Interactive & group Skills	-
Self-appraisal/Reflection on Practice	5
Planning and Management of Learning	1-4
Problem Solving	1, 2, 3
Communication & Presentation	4
Other skills (please specify)	-

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. The primary activity will be student-centred and will involve practical self-managed practical work .

**Assessment methods weightings which enable students to demonstrate the learning outcomes of the Unit:**

Program and documentation: 100%

**Indicative Reading for this Unit:**

**Main text**

Programming manuals

**Alternative Texts & Further Reading:**

**Guideline for Teaching and Learning Time (10 hours per credit)**

Lectures / Seminars / Tutorials / Workshops: 50 hours  
Tutorial support includes feedback on assignments and may vary by college according to local needs and wishes.

Directed learning: 50 hours  
Advance reading and preparation / Class preparation / Background reading / Group study / Portfolio / Diary etc

Self managed learning: 100 hours  
Working through the given real-life problem and designing and developing a working solution will take up the bulk of the learning time. In addition students are expected to engage with the tutor and other students and to undertake further reading using the web and/or libraries.

---

**Overview**

There is no examination for this module. It is tested totally by submitting documentation. The documentation should include all 18 sections listed below. These have all been defined more clearly to assist the candidates and centres in knowing what is required. Candidates must be made aware that the nature of this documentation makes it almost IMPOSSIBLE TO WRITE IT UP AFTER ALL IS COMPLETED. For instance, the testing process requires FULL PRINTED EVIDENCE and unless testing is ongoing and systematic, it will be difficult to produce full documentation later. THIS DOCUMENT should be covered in detail as part of class-work and be given to each candidate when DEVELOPING the programs.

The aim of the project is to test a candidate's ability to apply programming skills in the development of a REAL situation (normally related to business) and extend him/herself beyond the classroom programming exercise stage.

The project must produce a working solution to the problem. This problem should be of a NON-TRIVIAL nature; avoid projects like payroll which can be very involved. Past projects on payroll have been naïve simplifications, unrealistic and unworkable in the real world where a commercial

payroll package may have hundreds of different possible and complex adjustments to an employee's monthly salary.

ALL CANDIDATES AT A CENTRE MUST CHOOSE DIFFERENT PROJECTS.

Any appropriate computer language may be used. However, if a method is adopted where code is generated AUTOMATICALLY, then a listing of this code MUST be included and it must be FULLY ANNOTATED showing which part of the coding deals with which aspects of the project. THIS IS A PROGRAMMING PROJECT, NOT A DATABASE PROJECT.

The submitted project work MUST contain all the documentation relating to the production and testing of the programs. In particular, items 12 to 15 below are all components of the testing process and each should show how it relates to particular parts of the other sections. Previous submissions by candidates have included either no testing or minimal. For instance, if the project is about a customer ordering system, it is not satisfactory to have one or two customers and one or two different products – this does not test the accessing of files adequately nor does it show realistic billing which could spill over onto a second page. The tests MUST also be put into context. e.g. For a given test, manually write on the printout to which test in the test plan it refers, which set of test data was used to produce it and to which test it relates in the test log. Where output is on the screen, screen prints should be provided. Only if these sections are handled correctly is the candidate likely to score highly.

### 1.1 Sequence of events

1. **Start of Course.** Select a suitable project to solve. It should really contain at least TWO data files. All candidates at a given centre must choose DIFFERENT projects.
2. **Read the documentation requirements.** It is essential that the candidate knows what is expected BEFORE any significant work is started.
3. **In the first weeks of the course.** Complete an ICM approval form and submit it to the Institute. ICM will return this quickly with comments. The aim of this is to ensure that the project is neither TOO simple nor TOO difficult. If ICM considers it too difficult and would involve too much time and effort, it will either advise changing or suggest that only a part of the project be completed. Some previous projects have been so trivial they could not pass however good the documentation might be.
4. **During the course.** Solve the problem through a systematic development method, test the work and document it throughout. NOTE. There must be printed evidence of full testing. See items 12 to 15 in the list below.
5. **End of course.** Check that you have documented ALL elements from the attached list. This list has been detailed more fully to clarify the requirements more precisely. Put your work into order – preferably the order of the list and submit to the Institute. DO NOT SEND DISKETTES.

### 1.2 Documentation

Marks	Explanation
1. 5	<b>Definition of the problem and limitations of the programs.</b> ALL programs have limitations and never completely solve the problem – identify these. Explain WHY the program is being written and for whom. DO NOT include a list of thanks to tutors/friends etc - this has nothing to do with the running of the program, wastes time and will be ignored.
2. 5	<b>Objectives of the program.</b> This is not asking for AIMS. Objectives are <u>measurable</u> . A program either produces an invoice or it does not. AIMS such as “to make the data entry easy for the user” are not measurable and therefore should not be included.
3. 10	<b>Analysis of the current situation.</b> Explain in detail how the current system works. Include its shortcomings and any additional user needs.

4. 5 **Alternative solutions.** Identify alternative ways in which the problem could be solved. Some alternative ways are listed below but they need to be fully explained and justified:
- i buy a ready written package (if it exists – so you must identify it). Why write payroll programs when a package can be bought cheaply and working within hours? This is almost certainly the cheapest method and so there must be good reasons to reject it.
  - ii develop using a database system.
  - iii develop in one of several languages.
- State why your chosen method IS the most appropriate and why the others are rejected.
5. 10 **Data input.** List in data entry order, the specifications of each data set. It is useful to give each data set a meaningful name – e.g. 1. customer registration data. Data should be treated as a group of fields rather than many single items – e.g. all data for customer registration data or all data for an order item. For each field in each data set, include (i) fieldname, (ii) data type, (iii) any restrictions on it and (iv) full validation. A standard form could be devised.
6. 10 **Output.** Show a sample of every type of output whether on screen or on the printer. If the project is over-complex, this could present a real problem.
7. 10 **Files.** Identify all files involved. For each file state:
- (i) filename (ii) method or organisation (iii) method of access – there may be more than one access method depending on which aspect of the processing is used – differentiate between these clearly (iv) fieldnames (v) datatypes of each field (vi) field size. These last three items can best be tabulated in a standard form.
8. 15 **Process specification.** Describe ALL the processes in turn. For each state:
- (i) its purpose (ii) name the data input and output produced (iii) files involved and how each are accessed for THIS process (iv) where this process occurs in relation to others – e.g. after A and before C where A and C are named processes.
9. 20 **Algorithm.** This must use a structured approach. More emphasis will be placed on this section for projects where some of the code is produced automatically by the system. Pseudo-code is preferred.
10. 20 **Annotated structured listing.** A full listing of the code should be given. It should follow the algorithm CLOSELY. The code should contain the programmer's notes to explain how a particular feature is performed. Hand written comments should be added if these notes are incomplete. Program syntax must be totally correct. Any syntax errors will invalidate any testing applicable to that section of code and testing marks WILL BE REDUCED.
11. 5 **Variables list.** This must include EVERY variable used in the program except those input from fields of a file. Each must show
- (i) which module/section uses that variable, (ii) variable type (iii) size (iv) why it is needed. AVOID statements like "To hold customer number". Every variable "holds" something and in this case, WHY does the program need to have the customer number?- this is the reason for the variable's existence. There should be some system in your list – list either by program module/section (this may result in duplication) or perhaps alphabetical (but this must then show which module(s) use the variable).
12. 5 **Test Plan.** This is an outline showing the order in which processes/sections of code are tested. For instance ADD A NEW CUSTOMER should be fully tested before attempting any orders for that customer, amending customer details and certainly before deleting a customer record. Number each proposed test for easy cross-referencing.
13. 15 **Test Data.** This MUST be comprehensive. Unless a significant number of data records are on file, the testing will not show how a particular record is located or allow for multiple items such as a customer ordering many different products.

For each set of data, explain WHY it is present. e.g. If discounts are offered, then there must clearly be data to test for a customer who earns (i) NO discount, (ii) discount at one level (iii) discount at higher levels. In addition the test data must ensure that discounting does kick in at the right point by having test data AT the changeover point and immediately below it. Cross-reference each set to the Test Plan.

Testing must be FULLY documented. No documented testing will lose ALL the marks allocated for sections 12,13, 14 and 15 – nearly 25% of the project.

14. 20 **Testing.** Include printed evidence for ALL significant tests (see the notes for section 15). For each output, write in by hand, (i) a brief explanation of the purpose of the test (ii) cross reference to the test plan (iii) mark/comment on printouts of all items which are correct and which were not previously tested (iv) mark/comment on all items which are incorrect. For screen outputs, produce screen prints. For instance, as a guide, to show that a new customer has been added to a file, produce a suitable report BEFORE the new customer showing the customer details NOT present and the same print after the new customer has been added – highlight the differences by hand with a brief comment.
15. 5 **Log of Testing.** After each test, record in table form, (i) date and time (ii) the test being performed (cross-referenced to the Test Plan) (iii) the outcome of the test – either faults or processing assessed as being correct (iv) first thoughts about the remedy for correcting any observed fault.  
NOTE: It is not wrong to have tests which fail – all professional programmers make mistakes or overlook some feature. It IS however wrong to ignore faults that occur. There is no need to record details of a test that is a complete failure if this test is immediately followed by one that corrects some of these problems. Past test logs seem to show the candidate as being a brilliant programmer - the whole program completed in three runs and no errors! Is this likely?
16. 25 **User manual.** This should ideally be a SELF CONTAINED SECTION of the documentation describing (i) the program in outline, (ii) its options and limitations, (iii) sample inputs and outputs (iv) errors/problems which could occur and how to rectify them (v) operating instructions. If any part of this user manual should ALSO be elsewhere in the whole documentation, it MUST be here but in its other locations, a cross reference to the page number in the user manual can be given.
17. 10 **Technical Documentation.** This is for the maintenance programmer and relates to (i) any special methods employed, (ii) formulae used, (iii) data structures used such as matrices/linked lists, (iv) comments about the use of files, (v) back up procedures.
18. 5 **Appraisals.** This can be in two forms. If the program has been written for another user, that user should give a CRITICAL appraisal of its use and any problems should be noted. If there is no available end-user, the candidate can perform a self-appraisal but it must be CRITICAL. Are (i) there any features which could be added later or improved further, (ii) any features still not working etc... This appraisal is about the program, not the process of developing it so do not include problems you had with machine access etc. A claim that the program works fully will be discounted if there is no testing to prove it. In short, writing a few sentences to fill in this section is pointless unless it says something meaningful.

TOTAL 200 marks which will be halved to produce a percentage.

**Overall Comment:**

The candidate will gain more credit and marks for an honest assessment which shows that problems/errors have occurred and the steps taken to put them right. This still applies if the submitted project still has some logic errors and therefore only some of the submitted project is completed. This recognises that the creation of programs is not an exact science but requires means of putting things right. A candidate who appears to claim that everything works first time is not showing recognition of this process. There is also no point including sections of documentation that state very little and are merely there to try to satisfy the examiner.

Note that the user manual should be self-contained. It need not be detachable but the pages, if given to the user, would be all that the user needed. Therefore, other parts of the documentation should cross-reference into the user manual.

**Forms:**

Forms can be used to describe many items in this documentation. This can either be created by the candidate or supplied by the centre itself. However, if a centre-supplied form is not as appropriate as it should be perhaps because it was designed for a slightly different purpose, then the candidate should adapt it or design his/her own form.

Suitable forms could be used for details of:

1. Whole project progress time chart to show when each stage is completed.
2. Input data – one for each named and different SET of data.
3. File contents – one for each file.
4. Process specification.
5. Variables list.
6. Test Plan.
7. Test data including the purpose of that data – one for each set of data.
8. Test log with space to cross-reference to other test related items.

**10.1.9 HARDWARE AND OPERATING SYSTEMS Syllabus****Unit code: HOS-0908****Level: Advanced Diploma****Credits: 20****Unit leader: PY****Pre-requisites:****Main Aims of the Unit:**

This unit will provide an overview of hardware systems including technical specifications of input and output devices, memory and storage. Students should be able to produce a simple “pseudo code” to illustrate how data is handled in the processor. Further details are provided on operating systems and their function.

**Main Topics of Study:****A. Memory**

1. Types of main memory. ROM, RAM, DRAM, SRAM, EPROM, EEPROM, Registers, Cache.
2. Survey of relative speeds and costs of memory from fast registers to slow secondary memory (tape). Read and write cycle of memory.
3. Technical characteristics of types of main memory.
4. Memory maps.
5. Address bus. Operation codes. Reduced Instruction Set Codes (RISC). Decoding addresses.

**B. CPU**

1. Von Neumann model of the stored program.
2. Developments on this model – Pipelining, Array Processors, Multiprocessors.
3. Component parts of the CPU.
4. BIOS.
5. Data bus and address bus in relation to other parts of the computer system.
6. Registers. Accumulator(s), MAR, MBR, CIR, PC. Register transfer language (Listed below).
7. Simplified instruction set. A subset is listed at the end of this syllabus.
8. Shift and Mask instructions used to extract/combine parts of data in a single location.
9. Addressing modes – Immediate, Direct, Indirect, Indexed.

**C. Fetch-Execute Cycle**

1. Overview of the cycle.
2. Taken in turn, how each of the basic registers are used throughout the fetch-execute cycle. Use of register transfer language to describe these operations.
3. Complete analysis of a 3-instruction assembly routine as it passes through the fetch execute cycle – the analysis to be expressed in register transfer language.

**D. Complex memory structures**

1. Stack. Limitations of stacks. PUSH and POP. Recursion – simple examples such as determining factorial 5 ( $5! = 5 \times 4 \times 3 \times 2 \times 1$ ) by a recursive method.
2. Queue. Holding a queue in memory – circular queue using an array. Algorithm to add or remove an item from a queue.
3. Linked lists. Algorithms to locate, delete or insert an item using a linked list.
4. Binary tree. Algorithms to locate, delete or insert an item using a binary tree.

**E. Input devices**

1. Survey of current input devices available – keyboard, concept keyboard, barcode reader, optical mark sense reader, optical character reader, magnetic ink character reader, digitiser.
2. Technical aspects of how each of the above devices read their inputs. This includes being able to locate the position of the data on the capture form. Candidates need to see samples.
3. Buffers. Double buffering.

**F. Output Devices**

1. Survey of the range of printers. Comparison of the features offered by each type of printer.
2. Technical aspects of how each type converts the data it receives into printed output.
3. VDU. Video RAM. Colour. Raster scanning. Graphics card.
4. Plotters - Flat bed, drum. Their operations.

**G. Storage devices and media**

1. Difference between device and medium. Role of the buffer in data transfer.
2. Survey of storage devices. Magnetic tape, cartridge and tape streamer. Diskette, hard disk, Winchester. USB pen drive. Range of optical devices – CD and DVD.
3. Definitions of storage terms – file, record, field, cylinder, track, sector, header label, inter-block gap. File allocation table (FAT).
4. How data is physically stored on a storage medium.
5. How each of the storage devices read and records data.
6. Data transfer checks. Parity and its purpose. Odd/even parity - be able to give numeric examples of parity checking showing acceptance/failure. Cyclic redundancy checks.

**H. Logic Devices and Boolean algebra**

1. AND, OR, NOT. Truth tables. Karnaugh maps.
2. Boolean algebra laws – associative, commutative, de Morgan's Theorem.
3. Simplification of Boolean expressions and proving identities using truth tables, Karnaugh maps, and algebraic manipulation.
4. Devise simple circuits for Boolean expressions. Half and Full Adders.
5. Conversion of expressions into NAND or NOR units.
6. Flip-flops and their uses.

**I. Operating System – (Note: Basic commands are covered in GENERAL PURPOSE SOFTWARE module).**

1. Range of operating systems now in use – DOS, Windows and UNIX. Means of control using GUI or commands.
2. Purpose of the operating system. General survey of features offered.
3. Interrupt handling. Priorities. Interrupt vector table.
4. Scheduling – high, medium and low level.
5. Memory management. Partitioning. Paging. Virtual memory.
6. File management.
7. Multiprogramming. Concurrent processing. Deadlock and locking.

**Learning Outcomes for the Unit:**

At the end of this Unit, students will be able to:

1. Explain how input and output devices work
2. Evaluate a range of storage devices and methods of data handling
3. Understand how internal logic is handled
4. Describe the primary functions of operating systems and how they are carried out.
5. Produce simple algorithms for data handling

The numbers below show which of the above module learning outcomes are related to particular cognitive and key skills:

Knowledge & Understanding 1-5  
Analysis 3, 5  
Synthesis/Creativity 5  
Evaluation 2, 4  
Interactive & group Skills -  
Self-appraisal/Reflection on Practice -  
Planning and Management of Learning -  
Problem Solving 4, 5  
Communication & Presentation -  
Other skills (please specify) -

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. The students are actively encouraged to form study groups to discuss course material which fosters a greater depth learning experience.

**Assessment methods weightings which enable students to demonstrate the learning outcomes of the Unit:**

3 hour examination: 100%  
(Answer any 5 questions from 8, each question representing 20% of the paper.)

**Main text**

Refer to the ICM website for notes on this subject

**Alternative Texts and Further Reading**

The Principles of Computer Hardware by A Clements – (Oxford University Press).  
ISBN 019 856454 6 (Third edition).

Computer Science for Advanced Level by R Bradley – (Stanley Thornes )  
ISBN 0 7487 4046 5 (Fourth edition).

Operating Systems incorporating UNIX and WINDOWS by Colin Ritchie (Letts)  
ISBN 1 85805 302 1 (Third Edition)

**Guideline for Teaching and Learning Time (10 hours per credit)**

Lectures / Seminars / Tutorials / Workshops: 50 hours  
Tutorial support includes feedback on assignments and may vary by college according to local needs and wishes.

Directed learning: 50 hours  
Advance reading and preparation / Class preparation / Background reading / Group study / Portfolio / Diary etc

Self managed learning: 100 hours

Working through the course text and completing assignments as required will take up the bulk of the learning time. In addition students are expected to engage with the tutor and other students and to undertake further reading using the web and/or libraries.

### Guidelines

- This module covers the TECHNICAL aspects of hardware. Candidates should be aware that user actions are likely to feature only as a MINOR part in any question. Tutors should consult technical books for details. It is not intended for a casual user of hardware devices. Two excellent books are listed below.
- Some notes are provided for a typical instruction set which could be used relating to the fetch-execute cycle. The instruction set only considers NUMERIC data. If candidates use any other instruction set, they will be required to define each instruction they use OR a copy of the instruction set must be included with EACH batch of entries for this module. See the list below.
- Some previous candidates found it difficult to describe exactly what happens at each step during a fetch-execute cycle. So, Register Transfer Language (RTL) has been introduced to enable operations within the fetch-execute to be described concisely and precisely. See the list below.
- Candidates should be able to produce an algorithm (pseudo-code is preferred) for manipulation of queues, linked-list and binary trees. This might best be achieved by providing students with one pseudo-code answer (e.g. add item to a linked list) to show the expected format and requiring the student to adapt it to one of the others.

### NOTES

#### 1. Register Transfer Language (RTL)

RTL is actually a notation rather than a language. There are four main elements of it.

[x]	Refers to the CONTENTS of register <u>x</u> .
M(a)	Refers to the CONTENTS of MEMORY with address <u>a</u>
[p] ← [q]	Transfer contents of register <u>q</u> to register <u>p</u> .
M(a) = 5	Memory with address <u>a</u> CONTAINS the value 5. (Statement, not a transfer)

Examples of its use:

- |    |                       |   |
|----|-----------------------|---|
| 1. | [PC] ← [PC] + 1       | Program counter incremented – e.g. after new instruction has been loaded from memory.   |
| 2. | [PC] ← [CIR(address)] | Address part of current instruction register is transferred to the PC after ALU has detected a TRUE condition for a BRANCH instruction.   |
| 3. | [CU] ← [CIR(op-code)] | Control unit receives the operation code part of the current instruction in order to decode it and determine how to execute this instruction.   |
| 4. | [MBR] ← [M(MAR)]      | The contents of memory whose address is currently in MAR is passed to MBR. This applies EITHER in the fetch phase where the next instruction must be collected from memory to be passed to CIR OR in the execution phase where data is collected from memory for processing in the ALU. |

All stages of the FETCH-EXECUTE CYCLE can be described by a series of RTL statements. The first few statements WILL be the same for each instruction but once the op-code has been decoded, the execute-phase will dictate different actions.

This same convention is used below to describe the operation of each of the assembler instructions.

2. **Assembler instruction set**

This is a simplified instruction set suitable for illustrating features covered by this module. IF candidates use the following set, they will NOT be expected to define the instructions they use to illustrate their answers. If another set is used, an explanation of each different operation code used MUST be defined as part of the answer. Only NUMERIC data will be considered. RTL is used to describe the operation of each op-code.

Assumptions:

1. Each instruction contains only ONE-ADDRESS.
2. The start of memory for an assembled program is:
 

Address 0	PC – program counter
Address 1	Single accumulator (acc) for use by the ALU.
3. Mnemonic addresses may be used. n can therefore be replaced by PRICE.
4. The single accumulator is implied in ALL instructions where it is applicable. Thus LDA PRICE means Load contents of Memory address PRICE into the accumulator.
5. ANY memory address can be used for indexed addressing. In which case, in the instruction set below showing Immediate Addressing, n will be replaced by n(M[m]) for Indexed Addressing. e.g. LDA PRICE(INDEX) Here PRICE effectively marks the start of a PRICE table (in consecutive memory locations) and INDEX marks which element of that table should be loaded into memory. INDEX will normally be set to zero to mark the start of the table (i.e. memory position PRICE + 0) and then incremented to move onto the next price in the table.

Instruction	Action	Nominal description	Address method
-------------	--------	---------------------	----------------

Instructions involving numbers into/out of accumulator

LDN	n	$[Acc] \leftarrow n$	Load number	(Immediate addressing)
LDA	n	$[Acc] \leftarrow [M(n)]$	Load contents	(Direct addressing)
ADN	n	$[Acc] \leftarrow [Acc] + n$	Add number	(Immediate)
ADD	n	$[Acc] \leftarrow [Acc] + [M(n)]$	Add contents	(Direct)
SUN	n	$[Acc] \leftarrow [Acc] - n$	Subtract number	(Immediate)
SUB	n	$[Acc] \leftarrow [Acc] - [M(n)]$	Subtract contents	(Direct)
MUN	n	$[Acc] \leftarrow [Acc] \times n$	Multiply by number	(Immediate)
MUL	n	$[Acc] \leftarrow [Acc] \times [M(n)]$	Multiply by contents	(Direct)
DIN	n	$[Acc] \leftarrow [Acc] / n$	Divide by number	(Immediate)
DIV	n	$[Acc] \leftarrow [Acc] / [M(n)]$	Divide by contents	(Direct)
STA	n	$[M(n)] \leftarrow [Acc]$	Store accumulator contents	(Direct)

Branching instructions

BRN	n	$[PC] \leftarrow [CIR(\text{address})]$	Unconditional branch	
BZE	n	$[PC] \leftarrow [CIR(\text{address})]$	ONLY IF $[Acc] = 0$	Conditional
branch				
BNZ	n	$[PC] \leftarrow [CIR(\text{address})]$	ONLY IF $[Acc] \neq 0$	Conditional
branch				
BGE	n	$[PC] \leftarrow [CIR(\text{address})]$	ONLY IF $[Acc] = 0$ OR $[Acc] > 0$	Conditional
branch				
BLE	n	$[PC] \leftarrow [CIR(\text{address})]$	ONLY IF $[Acc] = 0$ OR $[Acc] < 0$	Conditional
branch				
BGT	n	$[PC] \leftarrow [CIR(\text{address})]$	ONLY IF $[Acc] > 0$	Conditional
branch				
BLT	n	$[PC] \leftarrow [CIR(\text{address})]$	ONLY IF $[Acc] < 0$	Conditional
branch				

Input / Output

REA	n	[Acc] < {keyboard}	n has no function but must be present = Read
number			
WRI	n	[Screen] ← [Acc]	Using n character positions on the screen = Write
number			
WNL	n	[Screen]	Write n NEWLINES to screen.
Text data will not be considered			

Shift Instructions

SLL	n	[Acc] ← [Acc]	LOGICAL Shift n bits to the LEFT
SLR	n	[Acc] ← [Acc]	LOGICAL Shift n bits to the RIGHT
SAL	n	[Acc] ← [Acc]	ARITHMETIC Shift n bits to the LEFT
SAR	n	[Acc] ← [Acc]	ARITHMETIC Shift n bits to the RIGHT
SCL	n	[Acc] ← [Acc]	CIRCULAR Shift n bits to the LEFT
SCR	n	[Acc] ← [Acc]	CIRCULAR Shift n bits to the RIGHT

Mask Instructions

AND	n	[Acc] ← [Acc] AND [M(n)]	Only "1" in corresponding positions of Acc and memory carried forward.
OR	n	[Acc] ← [Acc] OR [M(n)]	"1" is carried forward in each bit if there is a "1" in either corresponding positions of Acc and memory.
NOT	n	[Acc] ← NOT [Acc]	ALL Bits of accumulator are inverted.

**10.1.10 MANAGING COMPUTER SYSTEMS Syllabus**

**Unit code: MCS-0908**

**Level: Advanced Diploma**

**Credits: 20**

**Unit leader: PY**

**Pre-requisites:**

**Main Aims of the Unit:**

This unit introduces organisational structures and some general management principles but concentrates on management in an IT/computing context. The unit considers roles of people in real business situations and how they are managed for effective processing. The unit also considers the management role in setting up appropriate security systems and monitoring them to ensure their effectiveness.

**Main Topics of Study:**

**A. Organisational Structures**

1. Hierarchical structures in business.
2. Matrix structures in business.
3. Management by objectives.
4. Centralisation of a business. Affects of decentralisation.

**B. General Management Principles**

1. Management control of a business.
2. Planning and scheduling.
3. Time management.
4. Reporting methods. Reporting to management.
5. Communication channels in business.
6. Motivation.

**C. Information**

1. Distinction between data and information.
2. Information at different levels – operational, tactical, strategic.
3. Presenting information – different methods.
4. What makes “good” information?
5. Form and report design.

**D. Business processes**

1. Forecasting.
2. Budgeting.
3. Decision systems. Executive information systems. Decision support systems. Expert systems.
4. IT aids for the manager.

**E. Network Analysis/PERT**

1. Network analysis as a means of project control.
2. Definitions of Activity, Event, Critical path, Float, Dummy activity. Activity resources.
3. Conventions for drawing a network diagram.
4. Draw a network from a table of activities.
5. Determine critical path and minimum project duration.
6. Gantt Chart. Draw a Gantt. Aggregating resources within a Gantt chart. Levelling resources over the duration of the project.
7. The role of the computer in project control.

**F. IT Personnel - The roles of IT personnel in the business - their job descriptions.**

1. Data Processing Manager, Computer Manager.
2. Network supervisor/controller.
3. Systems Analyst.
4. Programmer, Systems Programmer, Programmer Analyst.
5. End users and data clerks.

**G. Training and Recruiting**

1. Methods of recruiting particularly for IT staff.
2. Methods of selection particularly for IT staff.
3. Benefits/drawbacks of training existing staff against recruiting new already-trained staff.
4. Training needs. Individual training programs. Personnel development records (PDR).
5. Means of providing the right training.

**H. Security and Privacy of computer systems**

1. Difference between Security and Privacy.
2. Threats to the security of computer systems – hardware, software and data.
3. Means of minimising security problems.
4. Privacy issues in business data.
5. The manager's role in controlling security and privacy problems.

**I. The effects of computers on business**

1. Managing changes in business methods.
2. Changing role of personnel.
3. Changes which affect customers.
4. Health issues related to computer use.
5. Data Protection Acts and other laws relating to computer systems which affect management and staff.

**J. The small business**

1. Differences in running a small business compared with larger organisations.
2. Benefits of IT for the small business related to a PARTICULAR business.
3. Business plans.
4. Choosing the right computer system.
6. Problems facing the small business with regards to IT.
7. Leasing and Maintenance contracts.

**Learning Outcomes for the Unit:**

At the end of this Unit, students will be able to:

1. Describe different organisational structures and general management principles
2. Understand the manager's role in the operation of effective computer systems
3. Implement project management techniques and understand the role of key personnel
4. Evaluate the manager's role in changing systems

The numbers below show which of the above module learning outcomes are related to particular cognitive and key skills:

Knowledge & Understanding 1-4  
Analysis 3, 4  
Synthesis/Creativity 3  
Evaluation 4  
Interactive & group Skills -  
Self-appraisal/Reflection on Practice -  
Planning and Management of Learning 3  
Problem Solving 3, 4  
Communication & Presentation 2  
Other skills (please specify) -

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. The students are actively encouraged to form study groups to discuss course material which fosters a greater depth learning experience.

**Assessment methods weightings which enable students to demonstrate the learning outcomes of the Unit:**

3 hour examination: 100%  
(Answer 5 questions from 8, each question representing 20% of the paper)

**Indicative Reading for this Unit:**

**Main text**

Refer to the ICM website for appropriate learning material

**Alternative Texts & Further Reading:**

Management Theory and Practice by G A Cole (DPP) ISBN 1 85805 166 5 (Fifth edition)

Quantitative Techniques by T.Lucey (DPP) ISBN 1 84480106 3 (Sixth Edition)

Management Information Systems by T Lucey (Letts) ISBN 1 855805 303 X (Eighth edition)

**Guideline for Teaching and Learning Time (10 hours per credit)**

Lectures / Seminars / Tutorials / Workshops: 50 hours  
Tutorial support includes feedback on assignments and may vary by college according to local needs and wishes.

Directed learning: 50 hours  
Advance reading and preparation / Class preparation / Background reading / Group study / Portfolio / Diary etc

Self managed learning: 100 hours  
Working through the course text and completing assignments as required will take up the bulk of the learning time. In addition students are expected to engage with the tutor and other students and to undertake further reading using the web and/or libraries.

### Guidelines

- This module covers some general management principles but mainly tests aspects of management where computers are involved. Answers to questions generally should take the approach of the manager looking on rather than the staff performing tasks.
- Students should be introduced to the roles of people in real business situations and how they are managed for effective processing. Make every effort to relate people management to the real world.
- **The module contains a section on security and privacy. The emphasis must be on the management role in setting up appropriate security systems and monitoring them to ensure their effectiveness. A list of problems without REALISTIC solutions is unlikely to satisfy a question relating security of computer systems.**
- Similarly, in recruiting and training, the emphasis must be on the special problems of handling IT staff where the needs change so rapidly. Sending an employee off to a general college on a lengthy standard IT course is unlikely to be a solution to new needs in almost all situations. Distinguish carefully on WHAT training is needed (content) and HOW it will be delivered (WHERE, WHEN and by WHOM). Candidates frequently confuse these two aspects of training. Giving trainees a newspaper is hardly likely to be productive but this often appears.
- Approaching the teaching through case studies with student involvement is likely to produce more success than a traditional teaching approach.

**10.1.11 Networks Syllabus**

**Unit code: NK-0908**

**Level: Advanced Diploma**

**Credits: 20**

**Unit leader: PY**

**Pre-requisites:**

**Main Aims of the Unit:**

This unit provides an overview of different types of networks and their use in business systems. The importance of the internet and the use of emails and associated applications will be explored. Further details will be included on the development tools associated with web design and the associated security issues.

**Main Topics of Study:**

**A. Types of Networks**

1. Definitions of LAN, WAN and the distinction between them.
2. Advantages and disadvantages of networking in general compared with stand-alone systems.
3. Advantages and disadvantages of networking in a particular situation.
4. Definition of Bandwidth. Importance of bandwidth.
5. Open Systems Interconnection (OSI).

**B. LAN**

1. Network topologies – Star, Bus, Ring, peer-peer.
2. Advantages and disadvantages of each of the main types of network.
3. Factors affecting speeds of networks.
4. How signals are passed and controlled in each of the network types.
5. The client-server.

**C. WAN**

1. Analogue and digital signals. Roles of the modem and multiplexer. Time Division Multiplexing (TDM). Frequency Division Multiplexing (FDM). ISDN.
2. Polling individual workstations.
3. Switching techniques – circuit, message, packet.
4. EDI. Uses made by business. Advantages compared with traditional communications.
5. EFT. Uses made by business. Advantages compared with traditional payment methods.
6. ATM usage by banks. The sequence of events WITHIN THE COMPUTER for a successful transaction in TECHNICAL terms.

**D. Internet**

1. Its origins.
2. Business use of internet.
3. Uniform Resource Locators (URL). Domain name. Domain name suffix. IP address.
4. Components of the Internet. WWW. E-mail, File Transfer Protocol Servers, News.
5. Use of internet by individuals.
6. Use of internet by business and organisations.
7. Setting up your own domain name.

8. Search engines. Structuring queries to eliminate irrelevant information.
9. Cookies.
10. Affects on society – good and bad.

**E. E-mail**

1. Registering for e-mail and logging on.
2. The e-mail system as seen by the user.
3. How to send and receive e-mail. Attachments.
4. Options open to the user on receiving an e-mail message.

**F. Hypertext Markup Language (HTML) and JavaScript**

1. Purpose of the HTML language.
2. Format of the HTML language.
3. Simple web pages in HTML.
4. Javascript. Applets. Simple examples.

**G. Data transfers**

1. Hypertext Transfer Protocol (HTTP).
2. File transfer Protocol (FTP).
3. Transmission Control Protocol (TCP)
4. JPEG, MPEG formats

**H. Control of a network and its security**

1. Role of the network supervisor/manager.
2. Password systems. Allocation of passwords. Levels of access.
3. Network threats. Viruses and their control.
4. Hacking. Firewall.
5. Encryption. Public key encryption.

**Learning Outcomes for the Unit:**

At the end of this Unit, students will be able to:

1. Distinguish between different types of network (eg LAN and WAN)
2. Describe in detail the business uses of networks
3. Describe techniques and simple procedures used in the design of web pages
4. Evaluate security and privacy issues in relation to using networks

The numbers below show which of the above module learning outcomes are related to particular cognitive and key skills:

Knowledge & Understanding 1-4

Analysis 2, 4

Synthesis/Creativity 4

Evaluation 2, 4

Interactive & group Skills -

Self-appraisal/Reflection on Practice -

Planning and Management of Learning -

Problem Solving 2,4

Communication & Presentation -

Other skills (please specify) -

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. The students are

actively encouraged to form study groups to discuss course material which fosters a greater depth learning experience.

**Assessment methods weightings which enable students to demonstrate the learning outcomes of the Unit:**

3 hour examination: 100%  
(Answer any 5 questions from 8, each question equals 20% of the marks)

**Indicative Reading for this Unit:**

**Main text**

Refer to the ICM website for learning material

**Alternative Texts & Further Reading:**

A Level Computing by PM Heathcote & S Langfield - Payne Galloway  
ISBN 1 904467 52 0 (Fifth edition)

Computer Science for Advanced Level by R Bradley – Stanley Thornes  
ISBN 0 7487 4046 5 (Fourth edition).

**Guideline for Teaching and Learning Time (10 hours per credit)**

Lectures / Seminars / Tutorials / Workshops: 50 hours  
Tutorial support includes feedback on assignments and may vary by college according to local needs and wishes.

Directed learning: 50 hours  
Advance reading and preparation / Class preparation / Background reading / Group study / Portfolio / Diary etc

Self managed learning: 100 hours  
Working through the course text and completing assignments as required will take up the bulk of the learning time. In addition students are expected to engage with the tutor and other students and to undertake further reading using the web and/or libraries.

---

**Guidelines**

- Candidates need to be precise in their explanations for this technical subject. Defining a LAN as local does not explain “how local” the networks extends.
- Practical experience is desirable in using a LAN. Candidates should have used e-mail and the internet. A visit to an organisation that uses the internet for business would also be beneficial.
- Clear distinction should also be made between user actions and actions by hardware/software – for instance in drawing money from a bank using an ATM.
- If the centre has its own website, some benefit could be gained if the creator/controller could discuss his/her work with students. Similarly, discussion with a LAN supervisor would be useful.
- Only a brief introduction to HTML and Javascript is required. The named books cover this adequately.

**10.1.12 Feasibility Study Syllabus**

**Unit code: FS-0908**

**Level: Advanced Diploma**

**Credits: 20**

**Unit leader: PY**

**Pre-requisites:**

**Main Aims of the Unit:**

Students undertaking this unit need to select a specific computer project. This is likely to relate to a particular local organisation or company and will be used to analyse and solve a given problem.

**Main Topics of Study:**

The Feasibility Study will include areas such as: Terms of reference, Details and problems of the current system, Study methodology, possible solutions & recommendations including a description of the costs and benefits.

**Learning Outcomes for the Unit:**

At the end of this Unit, students will be able to:

1. Implement techniques learnt in other units to formulate potential solutions
2. Integrate knowledge from a wide range of sources
3. Write a concise report addressing the key issues
4. Analyse a range of solutions to make informed recommendations
5. Manage their own time and resources in the completion of the report

The numbers below show which of the above module learning outcomes are related to particular cognitive and key skills:

Knowledge & Understanding 1, 2, 3, 4  
 Analysis 4  
 Synthesis/Creativity 2  
 Evaluation 3, 4  
 Interactive & group Skills -  
 Self-appraisal/Reflection on Practice 5  
 Planning and Management of Learning 5  
 Problem Solving 4  
 Communication & Presentation 3  
 Other skills (please specify) -

**Learning and teaching methods/strategies used to enable the achievement of learning outcomes:**

Learning takes place on a number of levels through lectures, class discussion including problem review and analysis. Formal lectures provide a foundation of information on which the student builds through directed learning and self managed learning outside of the class. The students are actively encouraged to form study groups to discuss course material which fosters a greater depth learning experience.

**Assessment methods weightings which enable students to demonstrate the learning outcomes of the Unit:**

Written report 100%  
(Note: 25% of the overall mark may be obtained from presenting the report to a local independent panel.)

**Indicative Reading for this Unit:**

**Main text**

Refer to the ICM website for appropriate learning material

**Alternative Texts & Further Reading:**

Material relating to the target organisation eg company reports

**Guideline for Teaching and Learning Time (10 hours per credit)**

Lectures / Seminars / Tutorials / Workshops: 50 hours  
Tutorial support includes feedback on project proposals and continuing supervision through the study.

Directed learning: 50 hours  
Advance reading and preparation / Class preparation / Background reading / Specific research and development of overall project

Self managed learning: 100 hours  
Working through the course text and completing the research methodology. In addition students are expected to engage with the tutor and other students and to ensure that the study is progressing satisfactorily.

---

**Overview**

---

There is no examination for this module. It is tested by

1. Candidate-submitted documentation. (75%)
2. Presentation at the centre of the candidate's findings to an assessment panel of at least two knowledgeable members of staff.
3. FULL report from the assessment panel of the quality of the presentation. (25%)

The candidate is required to perform an in-depth study for a new system and then produce a Feasibility Report. The report is of the type that would be presented to a Board of Directors for them to discuss whether to go ahead with the project, reject it or choose some middle option. The project must be REAL and not generalised such as the introduction of stock control with no reference to any particular organisation.

The new system should be

- EITHER the introduction of a computer system to replace an existing manual system
- OR upgrading an existing system to provide more and better features.

It is intended that the project should be one that a Board would be required to approve. This would therefore eliminate projects such as changing software from one brand to another UNLESS this made SIGNIFICANT improvements in the quantity and quality of new features and made

SIGNIFICANT differences to the whole organisation. In this latter case, these improvements must be fully justified.

The report from the assessment panel must be a CRITICAL report of the candidate's performance at the presentation. Some previous reports have been too brief and even incompatible with the candidate's written report: in these cases, the assessment is marked totally on the candidate's documentation.

ALL CANDIDATES AT A CENTRE MUST CHOOSE DIFFERENT PROJECTS.

### Sequence of events

1. **Start of Course.** Select a suitable project to solve. All students at a given centre MUST choose DIFFERENT projects. Ideally, the centre could give each candidate a brief which outlines the requirements for the study as a systems analyst would be given in business.
2. **Read the documentation.** The candidate must be given a full copy of these notes so that he/she knows precisely what is required.
3. **In the first weeks of the course.** Complete an ICM approval form and submit it to the Institute. Explain as fully as possible, the remit of this study. The ICM examiner cannot be expected to know about local conditions. ICM will return this form quickly with comments. The aim of this is to ensure that the project is appropriate - neither TOO simple nor TOO difficult. If ICM considers it too difficult and would involve too much time and effort, it will either advise changing or suggest that only a part of the project is completed. Some previous projects have been so trivial they could not pass however good the documentation.
4. **During the course.** Perform the investigation. Record your findings AS YOU GO ALONG – it will be very difficult to remember detail later on. Record details on any interviews you conduct with prospective users. Be sure to investigate the SCALE OF EVENTS. For instance for a mail order company -how many customers the company has, the number of orders received every month, the number of products it sells, the number of current staff currently involved with this application, the number of computer terminals required for the new scheme,..etc.
5. **End of course.** Check that you have documented ALL elements from the attached list. Prepare a presentation for the "Board" of about 20-30 minutes which you should deliver to a panel of assessors at the examination centre. Be prepared to answer questions about your project. At the presentation, hand in your project work to the panel including the original approval form on which all marks will be recorded. DO NOT INCLUDE DISKETTES. The panel will assess your performance and write a report to send with your work to ICM.

### FEASIBILITY STUDY REPORT

This should include:

1. A clear statement of the terms of reference for the study including the boundaries/limits.
2. Details of the current system.
3. Problems with the current system.
4. Details of the study undertaken – interviews, scale of the old and new systems.....etc
5. Possible solutions.
6. Recommendations. You could recommend a full system AND a cheaper partial system to allow the Board to decide which to implement.
7. Costs and benefits of the solutions you have provided in 5 above.

This report will be marked out of 75 by examiners at the Institute and will take into account the quality of your report, how well it is set out and how realistic are your solutions. Do not leave the examiner guessing about any aspect of the feasibility study.

### **PRESENTATION TO THE ASSESSMENT PANEL**

1. The panel should consist of at least two people with experience of computer systems.
2. The presentation should be as VISUAL and FACTUAL as possible and last for about 20-30 minutes. Summary notes can be given to each member of the panel – a copy of these should be included with your report and suitably labelled. The panel may ask you to explain some aspects in more detail.
3. The panel will make notes about your presentation, discuss these afterwards between themselves and then write a report that you will NOT see and which will go to the Institute with your documentation. The report will consider:
  - i Critical assessment of the overall quality of the presentation.
  - ii How confidently you delivered your presentation.
  - iii How detailed were your findings.
  - iv How realistic were your recommendations.
  - v How accurate were your figures.
  - vi An assessment of your conclusions.
  - vii An assessment of your cost/benefit findings.

The report will include brief details of EACH MEMBER OF THE ASSESSMENT PANEL including status in the organisation and past experiences.

NOTE: The assessment panel's report is to be marked out of 25. However, if it is too brief or inconsistent with the candidate's report, this centre mark will be ignored and the work will be assessed entirely by the Institute examiners on the candidate's written report.

### **Overall Comments:**

1. The candidate should be given a copy of these notes.
2. An honest report will score more highly than one that tries to hide problems or shortcomings in the investigations. These problems WILL be obvious to the examiner. Omissions will be taken to mean that part of the study was NOT undertaken or completed. Insufficient numbers will show that the scale of the problem was not considered - a real Board of Directors would quickly want more answers.
3. It is suggested that the presentation is marked as a percentage and the final mark divided by 4.

FEASIBILITY STUDY APPROVAL FORM

CENTRE

NAME.....

CANDIDATE NAME.....

CANDIDATE

NUMBER.....

PROJECT

TITLE.....

.....  
.....

1.2.1.1 NAME OF THE ORGANISATION FOR WHICH THIS STUDY APPLIES

1.2.1.2 SUMMARY OF THE PURPOSE OF THE STUDY (use additional sheet if necessary):

Supply sufficient information that the examiner will be able to judge to the depth and scope of study involved.

CANDIDATE

SIGNATURE.....DATE.....

Submit this form for approval AND also as the FRONT sheet of the final documentation.

EXAMINER COMMENTS and SUGGESTED ALTERATIONS:

Examiner.....

Date.....

MARKS AWARDED:

By Centre for the presentation: ..... out of 25

Detailed allocation shown in assessment panel report

ICM marking

Assessment panel report full and realistic	Assessment panel not acceptable
--	---------------------------------

ITEM	Maximum Mark	Mark Awarded		Maximum Mark	Mark Awarded
Terms of reference / Overview	10			10	
Description of existing system	10			15	
Problems with the existing system	5			10	
Investigation report – interviews etc	10			15	
Possible solutions	10			15	
Benefits of new system	5			10	
Costing for new system	10			10	
Recommendations	10			10	
Overall presentation of the work	5			5	
TOTAL	75			100	

ICM MARK	
CENTRE MARK	
TOTAL	
GRADE	