SYLLABUS CONTENTS Page:

QUALIFICATION STRUCTURE

The ACP qualification framework offers three professional levels:

- Certificate in Information Technology and Programming.
- Diploma in Information Systems Analysis and Design.
- Advanced Diploma in Computer Science.

The levels are normally taught on full-time courses (comprising each week 15 hours classroom tuition plus a minimum of 3 hours hands-on practical work). The Certificate and Diploma courses require a full academic year, typically 40 teaching weeks; the Advanced Diploma course is normally taught over a second academic year. The detailed content of each syllabus subject is shown in pages 3 to 11. Tutors are expected to teach current material which reflects developments in hardware, software and applications.

Each professional level has three written examinations and a practical component, all of which are compulsory. Each examination lasts two and a half hours and is conducted on a closed-book basis (candidates may not refer to notes or other reference material). Examination sessions for every subject are arranged three times each year as promulgated in ACP's examination calendar.

This syllabus should be read in conjunction with the ACP's brochure on Student Information and with ACP Centre Notes.

PRACTICAL COMPONENTS

Centres will need to provide adequate practical facilities in line with current syllabus requirements; these include personal computers, a network (for the Advanced Diploma), modern system software, approved programming languages, and up-to-date applications packages (for word processing, spreadsheets, graphics, and database assignments).

The practical requirements for each level are shown on pages 5, 8 and 11. At the Certificate and Advanced Diploma levels, full project documentation is expected; this can take the form of a bound A4 size folder or CD-ROM format. At Diploma level, two assignments for each subject are needed, each of some 1-2 days' work.

Any candidate's practical work may be requested by ACP for moderation. At the Advanced Diploma level, all project work must be externally moderated. Projects must be submitted on CD-ROM, and they should be supplied in a standard format. A class of student projects should be batched on a single, labelled CD-ROM. Each student's work should be stored in a separate folder (sub-directory). The folder should contain one document, preferably in *Microsoft Word*; this file should start with a detailed contents list (index) and should include any program files. The program files should be stored in the folder in text format with any sample data files. Projects will not be returned to candidates after moderation.

READING LIST

Suggested textbooks are listed within the detailed syllabuses. An especially recommended title is shown in bold type.

TUTORIAL MATERIAL

Current material available from ACP includes:

- Past question papers and specimen answers.
- Sample projects at the Advanced Diploma levels.
- Standard marksheet for programming projects and Applied Programming scripts.
- Chief Examiner's notes on the Applied Programming examination.

CERTIFICATE IN INFORMATION TECHNOLOGY AND PROGRAMMING

Candidates are required to obtain a Pass grade in three written papers, each carrying 100 marks and of two and a half hours duration, and a practical programming project (100 marks). In the first two written papers, (Information Technology Fundamentals and Computer Applications and Operations) there is a free choice of five questions from eight. In the Applied Programming paper, one question should be chosen from the three available.

INFORMATION TECHNOLOGY FUNDAMENTALS

F1 Computer Concepts: Ideas of Information, Information Processing and Data. The Data Processing Cycle. Examples of computer applications. Definition of Hardware; broad classes of computers (mainframe, mini and microcomputers) and networks. Computer programs. The computer as a programmable device. Classes of software (system and application). Programming languages: purpose, facilities and common examples.

F2 Computer Hardware: The Central Processing Unit (Control Unit, Arithmetic and Logic Unit, Main Memory). Peripherals. The organisation of a simple computer. The storage of programs and data. Data and Control paths in the computer (buses or highways). The Fetch-execute Cycle.

F3 Data - its Representation and Input: The Stages (collection, preparation, verification, input methods). Input Devices and Media. On-line and Off-line peripherals. Verification and Validation methods.

F4 Input Devices: Description of common input devices and media (such as keyboards, light pens, mice, magnetic stripe readers, punched media, magnetic and optical character recognition, mark readers...), including simple physical principles of operation and practical applications.

F5 Output Methods, Devices and Media: Description of Displays, Printers, Plotters and Computer Output on Microfilm, including simple physical principles of operation and applications.

F6 Computer Storage: Levels of storage: register, main and backing store. Units of storage (bytes and words) and capacities (Kbytes, Mbytes and Gbytes). Definition of Access Time. Principles of construction of magnetic tape drives, magnetic disc drives (floppy and hard drives), CD-ROM and DVD; recordable and rewritable compact discs: CD-R and CD-RW.

F7 The Binary System. Reasons for employing binary in a computer. The advantages and disadvantages of binary. The binary representation of numbers, characters and program instructions. Octal and Hexadecimal forms. Conversion between decimal, binary, octal and hexadecimal integers. Binary addition. Arithmetic overflow. Boolean logic. Simple AND, OR and NOT functions in two and three variables. Truth Tables. Half-adder and Full-adder logic. Logic diagrams.

F8 Programming Languages: Ideas of generations of programming languages: fourth generation (4GL), third generation ('high level'), assembly and binary machine code. Suitable applications for each level; comparisons between the levels. Translater programs - compilers, interpreters and assemblers; source code and object code. The concept of 'visual' languages. Java and the platform independence of its programs.

F9 Data Files: Definitions of file, record, field and character. The concepts of file organisation, file access and file processing (updating). The main types of data file such as master and transaction. Serial, sequential and indexed sequential organisation. Direct access and serial access. Updating sequential (tape or disc) files and indexed sequential files. Concepts of a simple database.

F10 Simple Telecommunications: Serial and Parallel transmission compared. Simplex, Half-duplex and Duplex modes. Modems and Multiplexors. Simple Interfaces. Character Codes. Basic communications facilities and the concept of bandwidth.

F11 Common Applications of Computer Systems: Non-technical descriptions (purpose, hardware, data, processes, outputs, advantages and limitations) in banking, education, engineering, police, hospitals, credit reference, meteorology, airline reservation and stock control.

F12 Networking and the Internet: Concepts of Local Area Networks, Wide Area Networks and the Internet. The World Wide Web: the concept, its uses and possible disadvantages. Internet Service Providers. Web pages: construction and access; the role of Hypertext Markup Language (HTML) and Java. The concept of electronic mail and its basic uses. The basic functions of browsers.

Recommended Textbooks:

Glossary of Computing Terms, British Computer Society; Longmans; ISBN 0582-36967-3 or ISBN 0582-47594-5
Computer Science by C S French, Fifth edition; Continuum; ISBN 0-8264-5460-7
Computing by Geoffrey Knott and Nick Waites, Third edition; Business Education Publishers; ISBN 1901-888215
Computers: Tools for an Information Age by Capron and Johnson, Eighth edition; Prentice Hall; ISBN 0-13-122723-8
Understanding Computer Science by Ray Bradley; Stanley Thornes; ISBN 0-7487-4046-5 (Tutorial and Student Use)

COMPUTER APPLICATIONS AND OPERATIONS

The emphasis is placed on the introduction, installation, operation and application of computers. Practical applications include databases, desk-top publishing, graphics, multimedia, networking, electronic communication, spreadsheets, and web pages.

CAO1 Problem solving with computers: a systematic approach. The role of software. A program as expressing a step-by-step solution to a problem (an 'algorithm'). The stages in programming: specification, analysis, devising the algorithm, flowcharting, programming, testing and documentation.

CAO2 Common computer processes: linear and binary searching, internal sorting, 2-way merging, sequential update, direct (on-line) update, calendar dates, times, validation routines, areas, volumes, ratios, percentages, integer division, switches, rogue values.

CAO3 Testing the logic of processes in flowchart, pseudocode or program form. Selection of comprehensive test data. Construction of effects table (dry run). Correcting faults and re-testing, including syntax and run-time errors.

CAO4 The features and facilities of a common third generation (high level) programming language. Simple ideas of program organisation, structure and style. Variable declarations. Elementary data structures: numeric, string, one and two dimensional numeric arrays, serial, sequential and direct access data files. Program constants, variables and their identifiers. Arithmetic relational and logical operators; rules of precedence. Expressions, assignment statements. Input and Output. Using control codes (escape sequences) to control peripheral devices.

CAO5 Control structures in programs: branching, loops, subroutines; nested structures. The use of common functions (such as integer part, type conversion, ASCII codes, string, sub-strings, random numbers) in practical applications.

CAO6 Program documentation. Detailed practical knowledge of specification, analysis and method, system outline, flowcharts (or alternatives), test plans, program listings (with appropriate comments), test results, input, output and file formats, user notes.

CAO7 Text Processing Package. Facilities and practical operation. Creating, saving and erasing text files. Entry and amending text. Cursor control: text insertion, over-typing, deletion. Operations on blocks of text: marking, moving and erasing. Simple formatting of text. Searching. Practical knowledge in the use of word processing software.

CAO8 Modes of computer operation: Batch and On-line (transaction processing and real-time systems) methods. Single user operations. Multiprogramming. Multi-access systems. The operating system and its various functions.

CAO9 A Simple Operating System. Facilities and practical operation. The organisation of a floppy diskette and its formatting. Directories and sub-directories. Copying and wildcard commands. Obtaining information about the system. Simple input and output. Error messages. Re-booting the system. Practical knowledge in the use of a common operating system such as *Windows 2000.*

CAO10 The work of computer operators. Comparison of work in small (desk-top) systems and mainframe computers. Distinction between the work of programmers and operators.

CAO11 The Computer Room. Equipment layout; environmental controls. Operating standards; discipline. Security considerations, including data back-up. The daily log.

CAO12 Hardware and data media. Routine operation of equipment. Precautions in handling magnetic media. Stationery: continuous, pre-printed, multi-part. Ancillary (off-line) equipment. Hardware faults: preventive and remedial maintenance. Consumables.

Recommended Textbooks:

Operating Systems, Third Edition, Colin Ritchie; Letts Educational; ISBN 1-85805-302-1
Computers: Tools for an Information Age by Capron and Johnson, Eighth edition; Prentice Hall; ISBN 0-13-122723-8
ECDL4 The Complete Coursebook for Microsoft Office 2000 by Munnelly and Holden; Pearson; ISBN 0-130-39915-9

APPLIED PROGRAMMING

AP1	Searching techniques: random and sequential data sets (one-dimensional arrays and files). Calculating and predicting the number of test accesses needed to locate a given record for different set sizes.
AP2	Sorting: well-known internal sort methods (such as exchange and insertion); counting the number of comparison and exchanges (or movements).
AP3	Merging: 2, 3 and 4 way merges of sequential arrays and data files. Refinements such as the elimination (or retention) of duplicated items.
AP4	Two-dimensional Arrays: the use of 2-dimensional arrays to represent and process practical applications (such as chessboards, aircraft seating plans, matrices).
AP5	Serial and Sequential File Processing. The record-by-record processing of serial data files to extract information or to update records. The updating of a sequential master file (on tape or disc) using a sorted transaction file; detecting and reporting common errors.
AP6	String Processing. Operations on (ASCII) character strings: length determination, searching for a character or sub-string, joining strings (concatenation). Examples in simple word-processing.
AP7	Simple Mathematics. Exercises in averages, ratios, percentages, areas, volumes, sets, finite series, factors and prime numbers. Integer arithmetic including quotient and remainder problems (such as fencing and tiling problems where unit sizes do not divide exactly into lengths or areas).
AP8	Currency Calculations. Financial problems involving simple interest, currency exchange rates and coin analysis.
AP9	Dates and Times. Calculation of interval between two dates or times. Relating weekdays and calendar dates. Extensions to leap years.
AP10	Computer Games and Random Numbers. The representation of simple games such as ‘noughts and crosses’. Games of chance with dice; the use of random number functions. Generating test or dummy data by means of random numbers.
AP11	Screen Control. Operations on alphanumeric screens: clear screen, position cursor, clear to end of screen, clear to end of line, clear a specified field. The design of a screen layout to facilitate data entry (with suitable captions), error messages and re-entry options.
AP12	Printer Control. Simple printer operations: output of text and control characters such as form feed, line feed, carriage return and backspace. Formatting a page (with title, column headings and record prints). Designing simple graphic output such as a histogram.

PROGRAMMING PROJECT

An individual programming project, written in an approved procedural language, such as QBasic, Java, Pascal,... will continue to be required. Languages such as Visual Basic or Access are not acceptable for projects at Certificate Level.

The subject for a Programming Project must be approved in advance to ensure it is suitable in terms of language, syllabus coverage, difficulty and work content. Typically, at least 50 hours’ work will be required to complete the project and its documentation. Refer to Centre Notes and Student Information for guidance.

DIPLOMA IN INFORMATION SYSTEMS ANALYSIS AND DESIGN

Candidates are required to obtain a Pass grade in three written papers, each carrying 100 marks and of two and a half hours duration, and a practical component comprising six individual assignments, two from each subject (totalling 100 marks). In each written paper, there is a free choice of five questions from eight.

SYSTEMS ANALYSIS AND DESIGN

SAD1	Systems Analysis: the scope, purpose and methods of systems analysis; the system life cycle and its stages; the roles of the participants, including user involvement. The systems analyst: qualities, qualifications and experience.
SAD2	Managing Systems Development: project planning and control - principles and methods such as Gantt charts and critical path networks. Justifying the new system. Quality assurance.
SAD3	Investigation and Analysis: initiation, terms of reference; investigation methods - records, interviews, questionnaires, observation, sampling; human factors and behaviour; techniques of analysis; recording facts; the full study report.
SAD4	Design and Specification: the system outline - output, input, files, processes; the human-computer interface; system flowcharts; computer run charts; procedure flowcharts; modes of operation.
SAD5	Output: definition of requirements - format, volume, peak loading; special stationery and media.
SAD6	Input: data capture methods; input specification - forms design and screen layout; the user dialogue; coding methods; error controls.
SAD7	Data Files: organisation; record structure; file sizes, growth and activity; access and processing requirements; choice of media.
SAD8	Documentation: needs; types and methods - text, charts, forms; standards.
SAD9	System Controls: physical and technical security; control of access; audit methods; legal requirements - simple ideas of data protection, copyright, intellectual property and computer misuse.
SAD10	Implementation: data conversion; staff training; changeover - methods, testing and acceptance; project review - compliance with specification, performance, financial benefit, development and support.
SAD11	Structured Systems Analysis and Design Methodology (SSADM): purpose and principles of SSADM; typical stages in a project; the tools of structured analysis; the separation of the logical and physical aspects; design controls.

Recommended Textbooks:

Systems Analysis and Design , by Yeates, Shields and Helmy; Pitman; ISBN 0-273-60066-4
Systems Analysis, Systems Design by Mason and Willcocks; Alfred Waller; ISBN 1-872474-09-8
Information Systems Development: Methodologies, Techniques and Tools by Avison and Fitzgerald, Third edition; McGraw-Hill; ISBN 0-07-709626-6 (Tutorial use)
The Information Systems Development Life Cycle by Avison and Shah; McGraw-Hill; ISBN 0-07-709244-9 (Tutorial use)

SOFTWARE ENGINEERING

Software engineering will be treated from a practical programming approach. Languages such as Visual Basic, Visual FoxPro or Access **and** either Java, C#, Pascal or 'C' should be introduced and employed for practical assignments.

SE1 Software: common deficiencies in software; the need for remedies and an effective method of managing software development. Software engineering methods. The stages in the software life-cycle - specification of requirements, design, review, detailed design, coding, testing, documentation and maintenance. The waterfall model; the spiral development model and simple prototyping.

SE2 Specification and Design Methodology: the properties of a sound specification. Reviews. Aims of a program design methodology. The style and presentation of programs. Detailed design and development methods - top-down design, decomposition, stepwise refinement. Modularity. The use of structure diagrams, flowcharts, pseudocode (program design language), decision tables and other methods. Candidates must be able to convert a simple logical problem into a specification.

SE3 Program Structures: the components and organisation of programs. Syntax and semantics. Data types. Declaration of data items. Operations on data. Control structures - sequence, selection and repetition. Functions and procedures (subroutines).

SE4 Data Structures: grouping of data items. One-dimensional and two-dimensional arrays, strings and records. Array subscripts; row and column operations. String and sub-string operations. Fixed length records.

SE5 File Organisation, Access and Processing: serial, sequential and indexed sequential operation. Accessing and updating records. Hash (scatter) storage. File media constraints.

SE6 Common Processes (algorithms): Simple processes on sorting, searching, merging and insertion. Simple random number techniques.

SE7 Errors and Testing: Types of errors in software - syntax, logical and run-time. Stages of testing - unit, integration, system and acceptance. White-box and black-box techniques. Construction of test data and test plans. Alpha and Beta testing. Reviews.

SE8 Programming Languages: four main generations - machine code, assembly, high level and application generator. Comparison of the main features and facilities. Factors influencing the choice of language for typical applications. Detailed knowledge of facilities and limitations of one common high level language. Special-purpose languages.

SE9 The Development Environment: the operating system, text editor, compiler, interpreter, assembler and debugging facilities - roles of each. Comparison between compilation and interpretation. Stages in compilation. One-pass and two-pass assemblers - operation. Debugging with single-step and trace facilities.

SE10 Fourth Generation Language: detailed working knowledge of a common database language. Record structure definition. File creation. Data entry and amendment. Screen formatting. Report generation. Indexing. Searching. Database commands and the development of programs.

Recommended Textbooks:

Software Engineering: A Programming Approach by Douglas Bell; Addison-Wesley; ISBN 0-20-264856-7
Software Engineering by Bell, Morray and Pugh; Prentice Hall; ISBN 0-13-832536-7
Software Engineering by Ian Sommerville; Addison-Wesley; ISBN 0-20-13981-5X (Tutorial use)

BUSINESS INFORMATION SYSTEMS

BIS1 Business Systems: organisation - objectives, structure and management. Functions common to businesses - administration, production, marketing and sales. The flow of information within an organisation. Levels of reporting and decision making.

BIS2 Information Processing Requirements: the use of computers. The automated office; word processing, desk-top publishing, accounts and payroll. Stock control, order processing, invoicing, spreadsheet and simple database applications. The wide availability of computer systems based on standard hardware and software.

BIS3 Hardware: the choices available in terms of floppy and hard discs, memory capacity, screens and printers. Communications equipment; supplementary interfaces ('add-in cards') to support bar-code readers, mice, scanners and fax facilities. Single and multi-user systems; local and wide area networks.

BIS4 Software: the sources - comparison of bespoke and packaged solutions. The distinction between application packages and application generators. The features in common word-processing, spreadsheet and database packages. Compatibility considerations and the use of integrated software packages.

BIS5 Requirements: identifying areas for possible computer support. Establishing present and possible future needs. Determining boundaries for the new system; taking consultants' advice. Stating the requirements with the emphasis on software functions; sizing the hardware. Issuing a specification.

BIS6 Selection: inviting proposals from suppliers. Comparison of responses. Shortlisting techniques. Selection criteria and the final choice of supplier. Negotiating the purchase and post-sales support. Contents of the contract with the supplier.

BIS7 Implementation: the practical steps of integrating a computer system within a small business. Installation of equipment in the office. Loading software and data files. Staff training and establishing new working practices. Live running and monitoring progress.

BIS8 Management Aspects: budgetary considerations - initial purchase and long-term costs. The responsibilities of staff. Practical methods of audit; data security. Hardware and software maintenance. System development.

BIS9 Common Applications: details of outputs, inputs, file organisation and processes of accounts packages (nominal, sales and purchase), payroll package and stock control.

BIS10 Spreadsheet Operation: detailed working knowledge of a common spreadsheet package. Creating, loading, saving and erasing a worksheet. The entry and editing of labels, names, values and formulae. Cursor control. Working with ranges. Re-calculation of results. Sorting, copying and printing operations.

BIS11 Case Studies: examples of small business systems to highlight the potential benefits, limitations, disadvantages, problems and pitfalls.

BIS12 Introduction to electronic commerce: the uses of the Internet and the world wide web for electronic mail, advertising and commercial transactions. The practical steps in setting-up simple e-mail facilities and web pages, both passive and active. Advantages, risks and problems presented by e-commerce. Intranets and Extranets.

Recommended Textbooks:

Business Information Technology by Geoffrey Elliott and Susan Starkings; Prentice Hall; ISBN 0-582-29802-4
Mastering Global Information Systems by William Buchanan; Macmillan; ISBN 0-333-68951-8

DIPLOMA ASSIGNMENTS

Each candidate must submit two assignments for each subject at Diploma Level (a total of six assignments). The subject for each Diploma assignment must be approved in advance to ensure it is suitable in terms of syllabus coverage, difficulty and work content. Typically, each assignment should require between one and two days' work (8 to 16 hours) from a candidate. Refer to Centre Notes and Student Information for guidance.

From January 2008, there must be a strong practical content in the Diploma assignments. At least one assignment should contain largely spreadsheet tasks, another should be devoted to database, and a third should contain programming tasks. Ideally, a graphics package will be used if flowcharts are included.

ADVANCED DIPLOMA IN COMPUTER SCIENCE

Candidates are required to obtain a Pass grade in three written papers, carrying 100 marks and of two and a half hours duration, and a practical component (100 marks) comprising an individual project and a programming project. In each written paper there are two sections; five questions must be chosen from ten; at least two questions must be selected from each section.

INFORMATION SYSTEMS PRINCIPLES AND NETWORKING

ISPN1 The Stored Program Computer. Components and organisation of a simple processor. Concepts of architecture, buses and interfaces. Examples of industry standards. The execution cycle. Interrupts and their handling.

ISPN2 Storage. Levels of storage. Concepts of access time and transfer rate. Comparison of construction, capacity, cost per byte, volatility. Main store organisation and uses. Backing storage: magnetic and optical devices. Organisation of discs: performance considerations, buffering, direct memory access, intelligent controllers.

ISPN3 Peripherals. Characteristics of common devices. Information transfer between devices. Buffering, protocols; serial and parallel transfers. Polling and interrupt techniques.

ISPN4 Data Transmission and Communication. Character codes; ASCII and Unicode. Simple methods of transmission and the associated hardware; asynchronous and synchronous techniques. Channel capacity. Error detection and correction.

ISPN5 Instruction Formats. Wordlength. Concept of an instruction set. Addressing modes. Zero, one, two and three address instructions. Variable length instructions. Reduced instruction set processors. Comparison between Pentium, Sparc and Java machines.

ISPN6 Operating Systems. Evolution. The functions of an operating system. The user interface. The management of processes, memory and files. Performance improvements. System start, including bootstrapping methods. Concurrent processes. Spooling. Scheduling. System security. Comparison between common operating systems such as MS-DOS, Windows, UNIX.

ISPN7 Networking. Concepts and terms. Hardware and software components of Local Area Networks (LANs). The functioning of a network: the OSI and 802 models. Comparison between the LAN and the Wide Area Network (WAN). Commercially available communications facilities. Common protocols in use on the Internet and the services they support.

ISPN8 Numeric and non-numeric Data. Binary representation of integers and fractions. Fixed-point and floating-point methods. Floating-point arithmetic; normalisation. Arithmetic errors; accuracy; methods of improving accuracy. Negative numbers: practical methods. Examples in two's complement form. The IEEE standards. The storage of graphical and multimedia data. Data compression techniques.

Recommended Textbooks:

Structured Computer Organisation, Fifth Ed., A S Tanenbaum, Prentice Hall; ISBN 0-13-148521-0
Operating Systems, Third Edition, Colin Ritchie; Letts Educational; ISBN 1-85805-302-1
Glossary of Computing Terms, British Computer Society; Longmans; ISBN 0582-36967-3 or ISBN 0582-47594-5
Understanding Computer Science by Ray Bradley; Stanley Thornes; ISBN 0-7487-1979-2.
Mastering Global Information Systems by William Buchanan; Macmillan; ISBN 0-333-68951-8
Network Essentials; Microsoft Press; (Previous edition) ISBN 1-55615-806-8

INFORMATION SYSTEMS ANALYSIS AND DESIGN; ADVANCED PROGRAMMING

Structured Systems Analysis and Design

ISAD1 Review of conventional systems analysis and design methodology in terms of the life cycle approach, the stages, techniques, limitations and weaknesses. Common problems associated with computer projects conducted under conventional methodologies and their underlying causes. General aims of information systems development methodologies. The emergence of structured methodologies in analysis and design.

ISAD2 SSADM version 4: concepts and organisation; a hierarchical and prescriptive methodology; the modules, stages and steps in SSADM; the task as a unit of work; the logical and physical models; the user view of SSADM; information system project activities which fall outside the scope of SSADM. Detailed knowledge of stages: Feasibility, Investigation of Current Environment, Business System Options, and Requirements Specification. Outline knowledge of stages: Technical System Options, Logical Design, and Physical Design. The benefits and limitations of SSADM.

ISAD3 Techniques employed in SSADM. General techniques such as fact finding, documentation, reviews, cross-referencing, and structure diagrams. Specific techniques in terms of purpose and practical application: data flow modelling by analysis of function, decomposition from the context level to more detailed data flow diagrams; symbology; logical data modelling by analysis of data, entities and relationships, the degree and optionality of relationships, building, validating and enhancing the logical data model; the data store/entity cross-reference, the process/entity cross-reference and the detection of anomalies; entity-event modelling in terms of its broad objectives; the three complementary views of an information system; relational data analysis and its role in enhancing the logical data model. Broad appreciation of other techniques in SSADM: requirements definition, dialogue design, business system options, function definition, technical system options.

ISAD4 Case Studies in Systems Analysis and Design. Practical application of techniques to libraries, stock control, order processing, invoicing, office administration, rental, reservation and ticketing systems.

Advanced Programming

ISAD5 Object-oriented Programming. Concepts and advantages; comparison with structured programming techniques. Classes, objects and methods (functions or procedures), inheritance, encapsulation; class libraries. Practical exercises to reinforce the theory.

ISAD6 Programming Languages. Detailed practical knowledge of Java (recommended), Pascal or 'C': types, declarations, assignments, flow control, functions, procedures, parameters, scope, recursion, input-output, files; simple graphical and animation techniques; event handling; programming for graphic user interfaces.

ISAD7 Data Structures and Algorithms: simple and compound data structures including arrays, stacks, queues, trees and lists; techniques for storing data structures in main memory and on backing store; the creation of algorithms for processes such as searching, sorting; stack and queue operations; tree processes such as insertion, and traversal. Linked list operations and uses of stacks and queues. Implications for process time and storage requirements as the data volume increases. The implementation of data structures in Java, Pascal or 'C'.

ISAD8 Database Systems. General database concepts. Practical analysis, design, implementation and development. Examples in SQL using Access, Visual FoxPro, SQL7, or Oracle.

Recommended Textbooks:

Practical SSADM Version 4+, Second Ed., Philip L Weaver; Pitman; ISBN 0-273-62675-2
Core Java by S C Horstmann and G Cornell; Sun Microsystems; ISBN 0-13-766-957-7
Fundamentals of SQL Programming, Mata-Toledo and Cushman; Schaum; ISBN 0-07-135953-2

INFORMATION SYSTEMS MANAGEMENT

Projects and Personnel

ISM1 Organisation. Data processing facilities within an organisation; types of system. Data processing departments: typical staffing structures.

ISM2 The Project Life Cycle. Stages in the life cycle from initiation to support and obsolescence. Levels of project management. Relationships between management, project team, suppliers and users. Project planning and control. System documentation.

ISM3 Financial Planning and Control. Estimating life cycle costs; the financial model. Cost-benefit analysis: appraisal methods such as discounted cash flow and net present value. Software estimates.

ISM4 Methodology and Quantitative Techniques. A systems development methodology such as PROMPT. Techniques for resource allocation, planning and scheduling: networks, linear programming. Simulation and modelling.

ISM5 Staff. Recruitment and training. Job descriptions. Standards and discipline. The legal framework of employment.

ISM6 External Services. The computer services industry: consultants and bureaux. Computer leasing.

Operations Management

ISM7 Installation and Commissioning. Accommodation requirements. Site planning; environmental services. Delivery and installation of equipment. Trials and acceptance. Performance monitoring and tuning. Reliability.

ISM8 Support. Hardware maintenance requirements; service agreements with contractors. Standby systems. Software maintenance needs; software support arrangements. System change procedures. Documentation and control.

ISM9 System Security. Resources at risk. Threats to security. Risk management. Security precautions: access control, technical counter-measures, personnel aspects. Data security: data control procedures, data back-up and recovery. Security and Privacy. Summary of legislation: the Data Protection Act 1984, the Copyright, Designs and Patents Act (1988) and the Computer Misuse Act (1990); the Data Protection Act 1998.

ISM10 Media, Consumables and Libraries. The range of media and consumables required. Stock control procedures; simple methods of inventory management: economic order quantities. Installation libraries: data, software and manuals on differing media; custody and control procedures.

Recommended Textbooks:

Quantitative Techniques by Terry Lucey; Thomson Learning; ISBN 0-826-45854-8
Practical PRINCE by C Bentley; NCC Blackwell; ISBN 1-85554-143-2
Strategic Management and Information Systems, Second Edition; Wendy Robson; Prentice Hall; ISBN 0-273-61591-2

From January 2008, there will be an increased emphasis on project management and a reduction in the service management aspects of the syllabus.

ADVANCED DIPLOMA PROJECT WORK

Students are required to submit two projects: firstly, a programming project in Java, Pascal or 'C' and secondly a project in an approved subject. Each project will carry 50 marks towards the total project component (100 marks). The subject for a project must be approved in advance to ensure it is suitable in terms of language, syllabus coverage, difficulty and work content. Each project can be expected to demand 60 to 80 hours' work by the candidate. Suggested programming projects in Java, complete with tutorial notes and marking scheme are available for use by centres and tutors. Advanced Diploma projects must be marked by approved examiners and must be moderated externally by ACP. Refer to Centre Notes and Student Information for guidance.

JAVA PROGRAMMING : TRAINING APPROACH

In a remarkably short time, Java has become a most popular programming language. There are many reasons for this explosive growth amongst programmers and employers: Java is fully object-oriented; it is ideal for both general programming and Internet applications; it has few of the idiosyncrasies of its older relations 'C' and 'C++'; it is a delightful language to use; its development kit is free on the Internet; it is easy to learn and to teach.

For these and many other reasons, both technical and educational, we recommend that Java is used widely in the Professional courses. Java can certainly be employed from Certificate level upwards, if a training centre so wishes. However, it is desirable that students are exposed to more than one programming language. Apart from Java, some practical experience in two or three other languages should be provided, such as QBasic, a visual language (perhaps FoxPro or Visual Basic) and a database language (Access or FoxPro would be suitable). If Java is not taught at a centre, then Pascal or 'C' are sound alternatives. In other words, there is flexibility of choice with regard to programming languages.

However, Java is the preferred choice and we are encouraging its use in several ways - by recommending its adoption, by creating single subject qualifications, and by offering tutorial material. We have pioneered the teaching of Java at our leading centres for three years and can provide an effective teaching strategy. This is helpful because many Java textbooks have complicated examples and appear difficult to the novice. By providing a range of simple teaching programs in our tutorials, we find students make rapid progress. There is much to learn, and our students also discover that sustained application over months is needed to achieve mastery.

JAVA PROGRAMMING CERTIFICATION

We have organised Java training into four levels as follows:

Level 1: Java Syntax and Fundamental Application Programming.

The Level 1 syllabus is shown below and is designed to match the Certificate Programming Problem syllabus. This means that Java Level 1 can be taught in two ways:

a. As the language for the Certificate level course covering the Programming Problem and the Programming Project components.

b. As an intensive full-time conversion course lasting five to ten days for students who have already achieved the Professional Certificate level qualification. (The exact duration of the training will depend partly on whether the student completes an optional project as part of the course.) On completion of this course, the student may qualify for the:

(1) Level 1 Certificate in Java Programming, and, optionally:

(2) Level 1 Certificate in the Java Programming Project.

Level 2: Java Object-oriented Programming.

This course covers object-oriented features of Java and is intended for the student who plans to become a professional programmer. The course may be taught within the Advanced Diploma or as a dedicated full-time course lasting five to eight days. In summary, it covers classes, objects and methods (functions or procedures), inheritance, encapsulation; class libraries. On completion of this course and a project, the student may qualify for the:

Level 2 Certificate in Java Programming.

Level 3: Java Graphical and Applet Programming.

This level covers simple graphical and animation techniques, images, printing, event handling, programming for graphic user interfaces, applets, and the construction of web pages for embedding applets. On completion of this course and a project, the student may qualify for the:

Level 3 Certificate in Java Programming.

Level 4: Java Business Applications Programming.

This course will use a productivity tool, such as Borland JBuilder, to construct useful business applications, including web interactions. On completion of this course and a project, the student may qualify for the:

Level 4 Certificate in Java Programming.

JAVA LEVEL 1 PROGRAMMING : OUTLINE SYLLABUS

On completion of this course, you will be able to:

1. Describe the main features of the Java language.
2. Provide examples of applications suitable for Java.
3. Explain the main methods of compiling and executing Java programs.
4. Create a Java source program using a simple text editor.
5. Compile a Java source program to form a class file.
6. Execute a Java class file.
7. Recognise and correct syntax, logic and run-time errors.
8. Test a Java program thoroughly by selecting appropriate conditions and test data
9. Write correct variable declarations, expressions and assignments using Java's primitive data types.
10. Declare String variables and employ selected String methods.
11. Correctly employ Java selection and repetition constructs.
12. Create Java code fragments which implement boolean expressions containing AND, OR and NOT operators.
13. Declare one-dimensional arrays and correctly apply array subscripts to simple applications.
14. Handle input and output data streams for keyboard, screen and file transfers.
15. Trap run-time errors and handle the exceptions appropriately.
16. Express the logic of simple applications in program design language (PDL), or an equivalent form, and develop correctly functioning Java implementations.
17. Complete a documented and tested programming project in Java given a specification.
18. Correctly employ the following Java keywords:

boolean	break	byte	case	catch
char	class	do	double	else
false	final	finally	float	for
if	import	int	long	new
null	package	public	short	static
switch	true	try	void	while

19. Use the following standard methods:

java.io.DataInputStream.readByte();
java.io.DataOutputStream.write();
java.io.EOFException();
java.io.FileNotFoundException();
java.io.IOException();
java.io.PrintStream.print() and println();
java.lang.ArithmeticException();
java.lang.ArrayIndexOutofBoundsException();
java.lang.String.charAt();
java.lang.String.equals();
java.lang.String.length();
java.lang.String.substring();
java.lang.String.toLowerCase();
java.lang.String.toUpperCase();
java.lang.String.trim();

20. Define the following Java and general programming terms:

applet	application	array element	arithmetic operator
ASCII	assembly language	assignment	binary machine language
block	braces	browser	bytecode
case sensitive	cast	compiler	construct
declaration	definite iteration	definition	expression
Internet	interpreter	Java virtual machine	just in time compiler
object code	logical operator	platform independence	post-conditioned loop
pre-conditioned loop	relational operator	servlet	subscript
type	Unicode	World Wide Web	

JAVA LEVEL 2 PROGRAMMING : OUTLINE SYLLABUS

On completion of this course, you will be able to:

1. Write a Java application program containing two or more simple methods.
2. Call one method from another method (which shares the same class or resides in a different class).
3. Construct both void and non-void methods.
4. Construct and use methods with parameters (arguments).
5. Distinguish between class (static) methods and instance methods.
6. Distinguish between class variables and method variables.
7. Explain the scope of variables declared in methods and classes.
8. Distinguish between nested and recursive method calls.
9. Explain the functioning of access modifiers public, private and protected.
10. Explain the concept of encapsulation and describe its benefits in object-oriented programming.
11. Explain how accessor and mutator methods can retrieve and process private data.
12. Describe the structure of a typical Java class from which objects can be instantiated.
13. Explain the role of constructor methods.
14. *Distinguish between the concepts of overriding and overloading.*
15. Describe the principle of inheritance and explain the relationship between parent classes (base classes or superclasses) and sub-classes (child classes).
16. Create an array of objects instantiated from one parent class.
17. Explain how parameters can be passed in an inheritance hierarchy.
18. *Distinguish between simple inheritance and multiple inheritance.*
19. Construct an application involving two or more levels of inheritance with parameter passing, accessor and mutator methods.
20. Compare structured programming and object-oriented programming in terms of their features and advantages.
21. Correctly employ the following Java keywords (including those in Java level 1):

abstract	extends	final	private	protected
public	return	super	this	

22. Use the following standard methods and constant:

    java.lang.Math.cos();
    java.lang.Math.random();
    java.lang.Math.sin();
    java.lang.Math.sqrt();
    java.lang.Math.tan();
    java.lang double Math.PI;

23. Define the following Java and general programming terms (including those in Java level 1):

accessor	actual parameter	argument	behaviour
constructor	encapsulation	formal parameter	global
identity	inheritance	instance	local
method	modifier	mutator	nesting
object	*overloading*	*overriding*	parameter
parent	polymorphism	recursion	scope
state	sub-class	superclass	value

(Items in italics will not formally be tested in an examination.)