P·N·L

THE POLYTECHNIC OF
NORTH LONDON

One year full-time and Two year part-time course for the Postgraduate Diploma in Computing

Course Handbook 1989

Department of Computing, Mathematics and Statistics

TABLE OF CONTENTS

1 RATIONALE, AIMS AND OBJECTIVES.

Rationale :

The PNL Postgraduate Diploma in Computing has been running successfully since 1977. The power and capability (per unit cost) of microelectronic and information processing hardware has increased rapidly during this time and is likely to continue to increase. This has had the effect of expanding the number and scope of applications of such hardware and the software systems built thereon. There is a consequent increase in the demand for skilled information processing professionals with knowledge and experience across a broad spectrum of specialisms. It is intended that the Postgraduate Diploma should continue contributing to the satisfaction of this demand - specifically in the field of software systems implementation and maintenance.

Aims:

1 To provide a postgraduate conversion course for graduates who possess a first degree (or equivalent qualification) in a subject area not including a substantial computing content.

2 To provide a thorough grounding in the design, implementation and use of state-of-the-art computer systems and software.

3 To emphasise, throughout, the importance of the concepts of correct, reliable, maintainable and robust software and, in general, of Software Engineering. This will be regarded as a major aim of the course.

4 To complement and form a broad background to 'in-service' courses. In-service courses provided by employers (directly or indirectly via hardware/software vendors) are, in general, narrow and specific to particular hardware, packages, operating systems etc. Technical training and experience of this sort needs to be complemented by broader and deeper studies.

5 To develop in the student a critical awareness of new ideas, products and likely developments in the field of computing.

6 To give the student a firm foundation upon which further study of computing may be based.

7 To enhance the career opportunities of the student.

Objectives - To achieve all of the above aims and more generally the student should:

a) be able to analyse, design and construct reliable, robust computer systems conforming to a client specification

b) have a basic knowledge of several application packages and programming languages, and an in-depth knowledge of some of these

c) apply a wide range of software systems development tools

d) be able to enter higher degree courses, on passing the Postgraduate Diploma with distinction.

e) be well-prepared for work in the swiftly-moving computer industry and have the capacity to deal with such constant change

f) Teachers in schools and colleges should also find that the Diploma enables them to teach Computing up to Advanced Level or equivalent

2. COURSE STRUCTURE

The course is divided into 7 units as follows:

Unit 0: A Perspective on Computing
Unit 1: Principles of Computer Systems
Unit 2: Fundamentals of Structured Programming and Design
Unit 3: Analysis, Design and Development of Information Systems.
Unit 4: Software Implementation, Tools and Techniques
Unit 5: Option
Unit 6: Project

The Part-time course (PTE and PTD modes) extends over two full Academic years (i.e. September to September). The Full-time course extends over one full academic year.

The first two quarters of the course cover computer hardware, software techniques and capabilities (Unit 1), and basic program design and development skills (Unit 2).

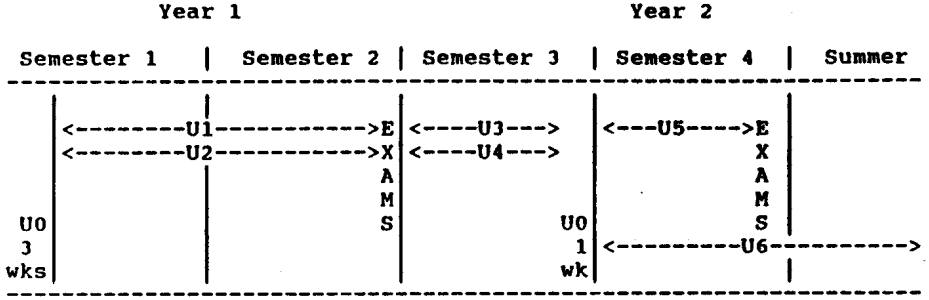
The third quarter of the course focuses on an integrated approach to the life-cycle of software systems and the concepts of Software Engineering (Units 3 and 4).

The final quarter of the course involves the study of a specialised option (Unit 5) and a substantial individual project (Unit 6).

In addition, Unit 0 is provided for all students in order to give a general perspective on computing and to introduce the student to some of the better-known types of application package.

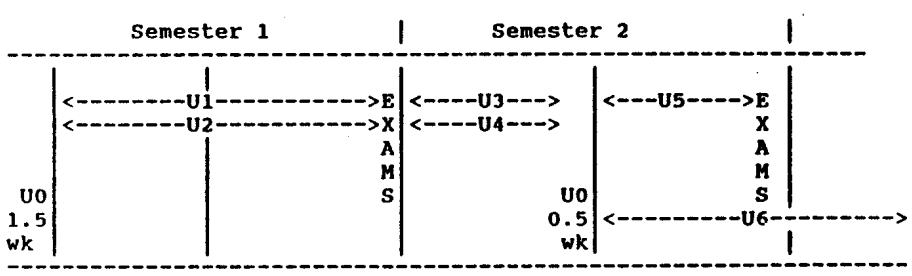
The Course structure is shown schematically in the following diagrams.

PTE and PTD Modes (2 Years)



Examinations for Unit 1 and Unit 2 in June of Year 1
Examinations for Unit 3, Unit 4 and Unit 5 + 0 in June of Year 2

Full-Time Mode (1 Year)



Examinations for Unit 1 and Unit 2 in February
Examinations for Unit 3, Unit 4 and Unit 5 + 0 in June

Examination Boards shall be held three times each year.

a) In February after the first semester examinations for students on the full-time course (Units 1 and 2)

b) In July, after the end-of-year examinations for full-time and part-time students (Units 1,2,3,4,5 + 0)

c) At the end of September, to approve the award of the Diploma once the results of the project are known (Unit 6).Referral results may also be dealt with at this examination board.

Duration of Units

Part-time Modes:

Unit	Hrs/Wk	Semester	No. of weeks	Total
0	6	1,3	4	24
1	3	1,2	27	81
2	3	1,2	27	81
3	3	3	14	42
4	3	3	14	42
5	3	4	15	45
6	3	4	15	45*
			Total	360

* plus project write-up time during the Summer vacation

Full-time mode:

Unit	Hrs/Wk	Semester	No. of weeks	Total
0	12	1,2	2	24
1	6	1	13.5	81
2	6	1	13.5	81
3	6	2	7	42
4	6	2	7	42
5	6	2	7.5	45
6	6	2	7.5	45*
			Total	360

*plus project write-up time during the Summer vacation

3 THE CURRICULUM

The first half of the course, which encompasses Units 1 and 2 is intended to provide the basic foundations of the core curriculum. It lasts one full academic year for the part-time modes and one semester for the full-time mode. Each Unit is examined at the end of this first half. The examinations form 50% of the assessment, the other 50% arising out of continuous assessment coursework.

Unit 2 is designed as a comprehensive programming course. Practical work will involve mainly programming assignments although other assignments which isolate one or another aspect of the development process (e.g. top-down design or testing) may be set. Unit 2 is complemented by Unit 1 which aims to provide a broad exposure to underlying computer systems and Unit 0 which aims to give students experience with some of the more commonly encountered microcomputer applications packages. Practical work for these Units will include simple systems programming (at the command file level) together with exercises in the use of the microcomputer packages.

Students who pass the first half of the course will be competent programmers capable of working on their own on small-scale programming tasks and well-versed in the rather ad-hoc development techniques appropriate for this type of work. At the same time they would be expected to have some insight into the complexity which arises as the scale of programming projects expands and to have sufficient grasp of the operation of the underlying computer systems to enable them to function effectively as users in a variety of computing environments.

The third quarter of the course encompasses Units 3 and 4. These are intended to run for the third semester for part-time mode students and are compressed into seven-and-a-half weeks for the full-time mode. This part endeavours to examine the conventional software life-cycle in some depth, making considerable use, where appropriate, of the automation aids and other techniques which are currently being developed in the field of Software Engineering.

Unit 3 tackles the early part of the life-cycle from project planning to system design. Using the same design notations (i.e documents which are the products of the system design phase), Unit 4 addresses the program design and implementation phases. At the same time, and in recognition of the "market value" of these topics, both Units 3 and 4 attempt to develop additional skills in certain areas. In the case of Unit 4, a second language, C, is taught, aimed at exploiting the acknowledged benefits of Unix as a well-established programming environment and as a vehicle for exploring the tools available in the Programmers' Workbench and in other proprietary support environments. Correspondingly, Unit 3 explores relational databases as a suitable mechanism for the development of certain types of Information Systems involving the use of application generation facilities and the standard query language (SQL).

Practical work in both of these Units will involve programming exercises in C and SQL respectively together with hands-on experience in the use of project management, analysis, design and implementation tools, under the auspices of an integrated support environment.

Students who pass this part of the course should be capable of making substantial technical contributions at a variety of levels to any team of computer professionals working on large-scale software projects. They should be conversant with the goals and operation of a representative sample of modern software production tools and should have some insight into the integration of these tools and the processes which they enable within the goals of the project as a whole.

Taken together, Units 0-4 represent a core curriculum. The purpose of Units 5 and 6 is to add a certain level of subject specialism tailored to the specific interests of individual or groups of students and to provide, via the project, a realistic programme of work within which to apply the principles they have learned and the skills which they have developed. All students will be offered a wide range of options (Unit 5) from which they should choose one. Each student also undertakes to do a project which may or may not be related to their choice of option. All students will have a project viva at the end of their final semester at which point the practical part of the project will normally be deemed to have been completed. However students will have the summer (with limited supervision and restricted access to computing facilities) during which to prepare their project reports.

By the end of the course students should have the basis for extending their studies into the specialist areas of their option and, through the project, should have had the opportunity to apply the skills and knowledge acquired on the course to suitable projects under conditions akin to working in an "industrial" environment - intensive development working to tight deadlines. At the same time, embodied in their project reports, they will have an objective record of their ability to apply themselves to a problem and to deliver a competent solution.

4 TEACHING, LEARNING AND ASSESSMENT METHODS

Unit Assessment Weightings

Unit	Weighting	
1	2	
2	2	
3	1	
4	1	
5	1	(includes Unit 0 assessment)
6	1	
Total	8	

Unit 0 material will be covered at various junctures throughout the taught course, and business/professional aspects of computing will be assessed in a compulsory section incorporated in the Unit 5 (option) paper. Practical sessions will give students the opportunity to become familiar with a variety of commonly used microcomputer applications packages.

Units 1 and 2 will consist of a series of lectures which progress the subject matter of the Units accompanied by practical work designed to reinforce skills, and techniques or to emphasize factual knowledge covered in the lectures.

Units 3 and 4 follow the same pattern except that there will be even heavier reliance on independent practical work to encompass the programming aspects and hands-on use of Software Engineering production tools. The course team will need to develop a series of case studies which reproduce the software development life-cycle so that the different phases do not need to be treated in isolation as they crop up in the course.

Unit 5 (the option) will be examined by both in-course assessment and examination.
Unit 6 (the project) will be examined by means of a viva at the end of the final semester, and by written project work to be submitted by the following September.

5 COURSE MANAGEMENT

Day to day management of the course is the responsibility of the Course Tutors (Principal and Modal) assisted by the Course Team. The Course Team comprises:

Head of the Computing Section
Course Tutors
Staff teaching on the course.

The Course Team, being appraised of objectives of the course, will ensure that an integrated approach is taken to the coverage of the syllabus material and the culminating project. Student involvement in the running and development of the course is ensured by the mechanism of Course Committees. Informal class meetings elect student representation to put forward their collective views to the formal Course Committees.

The terms of reference of Course Committees at PNL are as follows:

a) To make recommendations to the Faculty Board on matters of policy concerning the course.
b) To recommend to the Faculty Board formal changes to the scheme and/or regulations for the course.

c) To discuss formally matters generally affecting the course.
d) To monitor the progress of the course and report thereon to the Faculty Board.

Composition of the Course Committee

The Head of Department of Computing, Mathematics and Statistics.
Computing Section Leader.
Course Tutor for each mode of study.
All staff teaching the units on the course.
Two students from each class of the course* (i.e. two each from the first and second years of the part-time courses, and two from the full-time course).
Representative from PNLCS.
Representative from the Library.

*Where necessary, the voting strength of student members will be scaled to 50% of the staff numbers (i.e. 1/3 of the total votes).

6 COURSE REGULATIONS

A. Title of Course : Postgraduate Diploma in Computing
Award : PNL/CNAA Postgraduate Diploma

These regulations apply to students entering the course in 1988 and subsequent years.

B. Duration and Type of Course

PTE = 2 Year Part-Time Evening (2 evenings/week)
PTD = 2 Year Part-Time Day (1 day/week)
FT = 1 Year Full-Time

C. Minimum Entrance Requirements

i. The course is open to a wide range of students. The entrance qualifications are therefore flexible but students should normally have a first degree from a UK University or Polytechnic (or equivalent academic qualification).

ii. The Course Tutor may consider applicants without formal qualifications but with appropriate experience.

iii. Students will not normally be admitted to the Diploma if, in obtaining any previous formal qualification, they have covered a significant proportion of the subject matter of the Diploma course.

D. Admission and Selection Procedure

Application will be made on a special form obtainable from the Head of Department of Computing, Mathematics and Statistics. Wherever possible, applicants will be interviewed by the relevant Course Tutor, or a member of the course interview team, who may also ask an applicant to produce examples of his/her work.

E. Exemptions

There is no specific allowance for exemption from any of the Assessed units of the course.

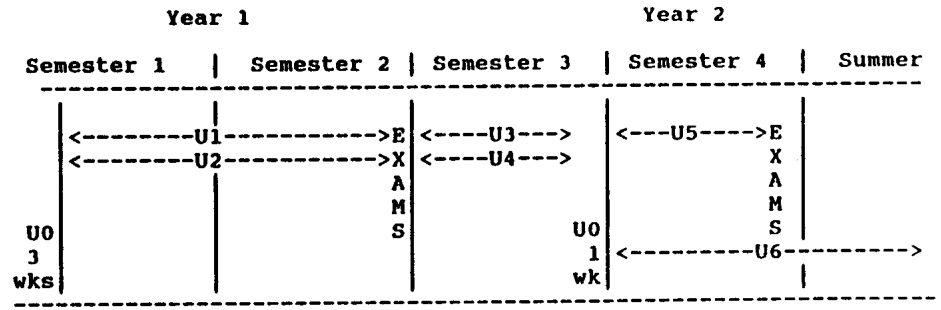
F. Course Requirements

The course is divided into 7 units as follows:

Unit 0 : A Perspective on Computing
Unit 1 : Principles of Computer Systems
Unit 2 : Fundamentals of Structured Program and Design
Unit 3 : Analysis, Design and Development of Information Systems.
Unit 4 : Software Implementation, Tools and Techniques
Unit 5 : Option
Unit 6 : Project

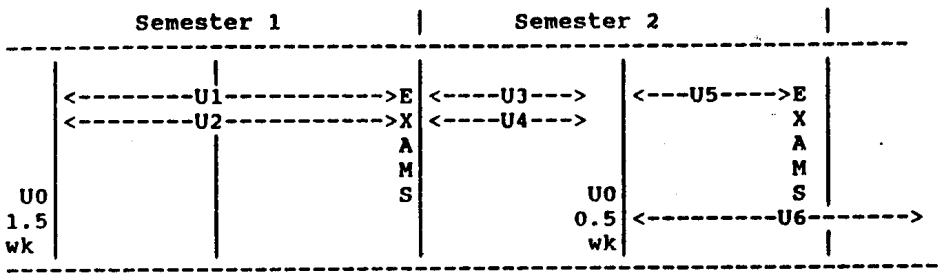
The Part-time Course (PTE and PTD modes) extends over two full Academic Years (i.e. September to September). The Full-Time Course extends over one full academic year.
The Course structure is shown schematically in the following diagram.

PTE and PTD Modes (2 Years)



Examinations for Unit 1 and Unit 2 in June of Year 1
Examinations for Unit 3, Unit 4 and Unit 5 + 0 in June of Year 2

Full-Time Mode (1 Year)



Examinations for Unit 1 and Unit 2 in February
Examinations for Unit 3, Unit 4 and Option in June

Examination Boards shall be held three times each year.

a) In February after the first semester examinations for students on the full-time course (Units 1 and 2)

b) In July, after the end-of-year examinations for full-time and part-time students (Units 1,2,3,4,5+0)

c) At the end of September, to approve the award of the Diploma once the results of the project are known (Unit 6). Referral results may also be dealt with at this examination board.

Duration of Units

Part-time Modes:

Unit	hrs/wk	semester	no.of weeks	total
0	6	1,3	4	24
1	3	1,2	27	81
2	3	1,2	27	81
3	3	3	14	42
4	3	3	14	42
5	3	4	15	45
6	3	4	15	45*
			Total	360

*plus project write-up time during the Summer vacation

Full-time mode:

Unit	hrs/wk	semester	no.of weeks	total
0	12	1,2	2	24
1	6	1	13.5	81
2	6	1	13.5	81
3	6	2	7	42
4	6	2	7	42
5	6	2	7.5	45
6	6	2	7.5	45*
			Total	360

*plus project write-up time during the Summer vacation

G. Assessment procedures

i The modes of assessment of each unit and the percentage o the total unit mark attributable to each mode of assessmen are as stated below:-

Unit	Coursework	Examination
1, 2, 3, 4,5	50%	50%
6	100%	

ii Assessment Weightings

The assessment weightings for each unit are shown below:

Unit	Weighting
1	2
2	2
3	1
4	1
5	1 (includes Unit 0 assessment)
6	1
Total	8

Examinations

iii Where the assessment of a unit includes a written examination this will consist of one unseen paper, normally 3 hours duration. The examinations will be held as follows:-

Mode	Units	Time
PTE & PTD	1, 2	June Year 1 (Semester 2)
	3, 4, 5	June Year 2 (Semester 4)
FT	1, 2,	February (Semester 1)
	3, 4, 5	June (Semester 2)

Coursework

iv Coursework exercises, practical work books and extended essays (hereinafter all called **coursework**) forming part of the assessment of a unit must be submitted by the appropriate dates decided by the Course Tutors. Students must be prepared to make their coursework available for inspection by the Board of Examiners if required. Coursework submitted late will normally be considered unsatisfactory.

v Where a candidate is at any time required to resubmit coursework, the Board of Examiners shall determine the date by which the coursework is to be resubmitted.

Definition of Unit Pass Mark

vi To pass a given unit a student must obtain an overall mark (i.e. weighted average mark) of at least 40%. Furthermore, the mark obtained in each mode of assessment must normally be at least 35% (i.e. 35% of the total mark available for that assessment mode).

Failed (a Unit) is defined as not passed (that Unit)

Progression

vii In order to progress to the second half of the course a student must normally obtain a pass in the two assessed units in the first half.

Procedure in the event of failure

viii A student may be required to retake a course or may be referred in an examination and/or be required to resubmit coursework as deemed appropriate to the individual case by the Board of Examiners.

ix A student may normally re-sit a written examination, or re-submit coursework once only. Referral examinations will take place at the end of the Spring term for full-time mode students and in September for part-time mode students.

Award of the Diploma

x In order to be awarded a Diploma a student must normally obtain:-

a. An overall average of 40% across all assessed units.

b. One, and only one, unit may be failed provided that:-

(1) that unit is not Unit 1 or Unit 2 or Unit 6
(2) that unit mark is not less than 35%.

Distinction

The Diploma, may at the discretion of the Board of Examiners, be awarded with Distinction to candidates of exceptional merit. To qualify for consideration for the award of a Distinction a candidate must normally obtain an overall average of at least 70%, provided that no assessment has been awarded less than 50%.

Time Constraint

xii A student will normally be expected to complete the Diploma course within four years of first registering in the case of the Part-Time modes and within three years of first registering in the case of the Full-Time mode. Any extension beyond this period will be at the discretion of the Board of Examiners.

Procedures in the event of illness or other sufficient cause

xiii If through illness or other cause deemed sufficient by the Board of Examiners, a student fails or fails to sit any written examination this may, on production of evidence acceptable to the Board, be condoned by the Board of Examiners if the Board is satisfied, by the students overall performance in the course, that they have sufficient evidence to arrive at a fair estimate of the appropriate mark for the unit(s) concerned.

xiv Where there is insufficient evidence, the Board of Examiners may permit the student to retake the examination in the appropriate unit(s). In the case of assessed units in the first half of the course, the Board of Examiners shall decide whether the student be permitted to proceed to the second half.

xv If through illness or other cause deemed sufficient by the Board of Examiners, a student fails or fails to submit coursework the student shall be permitted to resubmit (or submit for the first time as the case may be) that coursework which has been failed or missed. The coursework shall be resubmitted by a date to be determined by the Board of Examiners.

H. The Board of Examiners

i The Board of Examiners shall consist of:

The Head of Department of Computing, Mathematics and Statistics (Chairman)

The External Examiners

The Course Tutors of the course

Other staff teaching the assessed units of the Diploma

ii The duties of the Examining Board are set out in the following terms of reference:

1. To be responsible for all examination question papers, essay or project topics as required by the course regulations.

2. To make recommendations to the appropriate body for the award of qualifications to individual students.

3. To determine whether a student has complied with the requirements of the course and its assessment and may proceed to the following stage of the course.

4. To determine what action shall be taken in accordance with the provisions of the course regulations, in the case of a student failing or missing all or part of the assessment for any part of the course, or failing to comply with the requirements of the course and its assessment.

5. To determine, in accordance with the relevant Polytechnic regulations, what action shall be taken in the case of a student who has committed an examination offence.

6. To make recommendations to the relevant Course Committee(s) on any matters concerned with the assessment of the course.

7. To be responsible for such other matters as are attributed to it in the specific regulations of the course or may be referred to it by the Academic Board.

iii Examiners Meetings will normally be held three times a year, in February, July and September.

iv The result of an examination shall be determined at a meeting at which all members of the examining board are expected to be present, but of which the proceedings shall not be invalidated by the absence of a member or members of the Board through illness or other unavoidable cause provided that where applicable, at least one external examiner is present.

I. Appeals

Any appeal against the interpretation or implementation of any part of these regulations shall be made to the Academic Board in accordance with the procedure which may be determined by the Academic Board from time to time.

J. Amendments

These regulations may be amended by the Academic Board.

7. GENERAL INFORMATION

Notice Board :

We use the Postgraduate Diploma notice board (outside the Departmental Office) to inform students of general matters of information (such as timetable / room changes, examination timetables, coursework schedules etc.) and also, on occasions, to communicate with specific students. Please inspect this notice board regularly. This applies particularly to Part-Time and returning students.

Problems with fees and registration cards :

These are the responsibility of the SCT (= Science and Technology) Faculty Office (Tel Ext 2045) -- please address queries to them.

Change of address :

If you change your home or work address or telephone number(s) please inform both the Departmental and the SCT Faculty offices.

Letters confirming attendance on the course etc.:

Requests for these must be made in writing to the Course Tutor, with adequate prior warning.

Extenuating Circumstances Regarding Assessed Work :

see Section 9 "Provision of Written Evidence to Examination Boards"

Referred students :

Students who are "referred" by Exam boards (i.e. required to resit exams, resubmit courseworks, or retake whole units) must make it their responsibilty to make themselves aware of the exam dates, coursework deadlines &c. relevant to themselves; this particularly applies to students who are no longer attending the Polytechnic regularly; if they need extended Computing and/or Library access then they must pursue this with PNLCS/Library (in conjunction with their Course Tutor).

Employment references :

Please let your course tutor know that he/she is being quoted as a referee and, most importantly, the year(s) which you attended the Polytechnic and the year in which you gained the Diploma.

Graduation and Award of the Diploma :

The Graduation Ceremony and the award of the Diploma Scroll is organised and administered by the SCT Faculty Office; any queries should, initially, be addressed to them.

8. Syllabuses

Unit 0 : A Perspective on Computing

Aims

Although some students (particularly part-timers) have considerable work experience and contact with computers in their employment, many do not. This unit aims to broaden the general computing background of all students, helping them to relate their studies to organisational and social contexts, and to make more informed option choices. The specific aims are:

a) To introduce students to a broad range of computer packages and applications systems.

b) To encourage a critical approach to the evaluation of computer software and the social context of computing.

c) To introduce the organisational structures within which computing activities take place.

Syllabus

1. **Software Practice and Experience**

   - basic features of database packages
   - applications using spreadsheet packages
   - word processing and desktop publishing
   - software components of the automated office
   - interactive graphics systems

2. **The Computing Industry**

   - brief history of technical developments
   - hardware manufacturers and oems.
   - software houses.
   - user groups
   - publications

3. Industrial and Technical Applications

- CAD/CAM
- automated manufacturing
- process control
- embedded systems

4. Social Issues

- professional practice and codes of conduct
- data privacy and security, related legislation
- commercial and state and data usage
- employment issues
- change and obsolescence in post-industrial society

5. Business and Administrative Applications

- job roles and functions within an organisation
- major application areas.
- simple accounting concepts
- office automation
- centralised and distributed systems.
- non-commercial organisations

Indicative Practical Work

Produce a staff allocation table for the computing section using in-house Lotus software.

Produce reports on enrolment, fee-collection, and completion figures for the Department's short courses.

Run a newsletter for the Postgraduate Diploma using Departmental WP and DTP systems.

Produce diagrams and slides for seminars using in-house graphics packages.

Construct a bibliography on a specialist topic such as VLSI

Write a report on the personal data collection activities of the Polytechnic.

Give a seminar on changing employment patterns within students' workplaces (part-time course only).

Indicative Booklist

T. Forester	"The Information Technology Revolution" Basil Blackwell 1985
J. Shurkin	"Engines of the Mind: A History of the Computer" Norton 1984
R. Edwards	"Contested Terrain: The Transformation of the Workplace in the Twentieth Century" Heinemann 1979
T. Solomonides L.Levidow(eds.)	"Compulsive Technology: Computers as Culture" Free Association Books 1985
D. Burnham	"The Rise of the Computer State" Weidenfeld 1983
F. Blewett	"Microcomputers in Business" Newnes 1985
	"Mastering Financial Accounting" Macmillan Master Series
D.P.Curtin, L.P.Porter	"Microcomputers,Software and Applications" Prentice Hall, 1986

Manuals, user guides, and reports for packages and applications systems.

Journals, newsletters, and magazines published for the Computing industry and educational community.

Class Contact

24 hrs = 12 hrs lectures/seminars + 12 hrs tutorials/practicals

Assessment This unit is assessed in a compulsory section incorporated in the Unit 5 (Option) examination paper.

Unit 1: Principles of Computer Systems

Aims

To provide a basic understanding of the computer architecture and systems software of contemporary computer systems ranging from large-scale multi-user systems to single-user microcomputers.

To provide a systematic understanding of the facilities provided by hardware and systems software to the applications programmer.

To familiarise the student with current microcomputer technology and representative microcomputer applications packages.

Particular emphasis will be given to the hardware, software, systems and services available within the Polytechnic.

Syllabus

1. The Stored-Program Concept

- brief historical development, the steps involved in the solution of a problem by computer
- basic computer architecture; the function of and interaction between the central processor, memory, secondary storage and input/output devices

2. Representation and Storage of Data

- the bi-state representation of information, bits, bytes etc., binary and hexadecimal number systems
- the interpretation of bit patterns according to context (e.g. ASCII, instruction op-codes, unsigned integer)
- graphical representations
- storage devices, current storage device technology, access time, data transfer rates, comparative costs

3. Systems Software

- operating system functions from the user's viewpoint; command languages, file management, memory management
- the steps involved in taking a high-level language program from inception to a running program

4. Computer Architecture

- the use of a subset of a simple assembly language to illustrate processor architecture
- addressing techniques
- instruction codes
- I/O device control

5. Contemporary and Future Computer Equipment

- overall appreciation of the cost and performance of typical
- micro, mini and mainframe technologies
- a summary of peripheral devices available
- a survey of technologies currently under development
- procurement policies for hardware and software

6. An Introduction to Communications

- principles of serial and parallel data communications, highways and interfaces
- communication channels, baseband and broadband signalling,
- simplex/duplex operation, synchronous/asynchronous transmission
- protocols for information exchange
- file transfer
- local area networks.

Indicative Coursework

writing a radix conversion program
writing DCL command procedures to mimic MS-DOS commands
writing a simple assembly language module to link in to a high-level language program
writing arithmetic macros for multi-length integers

Indicative Booklist

P.Bishop	"Computing Science", Nelson, 1987 Nelson 1987
P. Norton	"Inside the IBM PC" Prentice Hall 1986
Ed. Strauss	"Inside the 80286" Prentice Hall 1986
P. Abel	"IBM PC Assembler Language and Programming" Prentice Hall, 1987.
J. Osbourne	"MS-DOS User's Guide" McGraw Hill 1985
L.J.Scanlon	"Assembly Language Programming with 8088/8086/80286" Prentice Hall 1988

Class Contact

81 hours = 54 hrs lectures + 27 hrs tutorials/practicals.

Assessment

Examination - 50%
Coursework - 50%

Unit 2: Fundamentals of Structured Programming and Design

Aims

To introduce the concepts of structured design and structured programming via a high level language such as Pascal or Modula-2.

To acquaint students with some of the established techniques used in program design and implementation.

To provide students with experience of designing, implementing, and documenting modularised systems in a team setting.

To introduce students to the concept of the System Life Cycle.

Emphasis will be placed on teamwork in the design and construction of small to medium sized systems. This naturally implies the production of clear structures at design and implementation stages.

Syllabus

1. **Introduction to Programming**

   - simple interactive programs
   - use of parameterless procedures
   - basic data types and control structures
   - stepwise refinement, use of pseudo-code
   - further data types
   - survey of fundamental algorithms

2. **Modular Programming**

   - procedures with parameters
   - problem solving using procedure abstraction
   - top down and bottom up implementation
   - documentation and testing
   - structure charts for design and documentation
   - creating and using external procedures/functions
   - teamwork to build a system as a library

3. **File Handling Techniques**

   - text files and filter programming
   - creating and using sequential record files
   - sorting and updating a master file
   - control break processing
   - interactive programming with ISAM files
   - direct access files and hashing algorithms

4. **The System Life Cycle**

   - Taught as two or three case studies
   - Students to work in teams
   - A small amount of practical work will be associated with each stage in the system life cycle

5. **Advanced Programming**

   - Dynamic data structures and applications
   - Recursion and the applicative approach to design
   - Heuristic search techniques
   - simulating decision tables
   - simulating control structures from other languages

6. **Programming Techniques**

   - Survey of available systems and applications libraries
   - Concepts of human-machine interfaces
   - Survey of advanced algorithms

Indicative Coursework

Specify and implement a library of file handling and data compression routines (e.g. Huffman codes). Use this library to produce some simple statistics about the performance of data compression methods on various sorts of data files.

Indicative Book List

P.D.Terry	"An Introduction to Programming with Modula2 Addison Wesley 1987
P.A.Messer and I. Marshall	"Modula2: Constructive Program Development" Blackwell 1986
A.J.Sale	"Modula2: Discipline and Design" Addison Wesley 1986
N.Wirth	"Algorithms and Data Structures" Prentice Hall 1986

Class Contact

81 hours = 54 hours lectures + 27 hours tutorials

Assessment

Examination - 50%
Coursework - 50%

Unit 3: Analysis, Design and Development of Information Systems

Aims

To acquaint students with current analysis techniques generally in use, but with special reference to Information Systems.

To acquaint students with current design notations arising out of those analysis techniques.

To provide students with practical experience in the use of automated tools and other techniques for producing and verifying design documentation.

To provide students with an introduction to databases with special reference to relational databases.

To provide students with experience in the use of the standard language SQL.

Parts A and B will be taught in parallel.

Syllabus

Part A

1. **Project Planning**

   - project organisation
   - estimation and cost modelling

2. **System Analysis**

   - data models: entity relationship models, subtyping
   - activity models: data flow, activity hierarchies

3. **System Design**

   - design notations and design documentation
   - design tools: CASE tools, data dictionaries
   - verification and prototyping: 4GL application generation (e.g. in INGRES), 5GL specification languages (e.g. Hope)

Part B

4. **Information Systems Development**

   - introduction to database theory
   - relational databases
   - programming with SQL

Indicative Coursework

Analyse and design a simple system using both data and activity modelling perspectives; implement that system using INGRES; use automation where appropriate to progress through the life-cycle of this system.

Indicative Booklist

C.J.Date	"An Introduction to Database Systems Vol 1" (Fourth Edition) Addison Wesley 1986
Tom de Marco	"Structured Analysis and System Specification" Yourdon 1979
R.Rock Evans	"Analysis within the System Development Life Cycle" PERGAMON Infotech 1987
C.J.Date	"A Guide to INGRES" Addison Wesley 1987
	"INGRES" manuals
	"Computerised Assistance during the Information Systems Life Cycle" Proceedings of CRIS 88: (North Holland) 1988

Class Contact

42 hours = 28 hour lectures + 14 hours tutorials

Assessment

50% Examination
50% Coursework

Unit 4: Software Implementation, Tools and Techniques

Aims

The purpose of this unit is to provide students with the expertise to work effectively on medium to large software projects from mid-design stage (i.e. comprehensive requirements specification) through detailed design, implementation, and maintenance stages. This is to be accomplished by meeting the following subsidiary aims:

a) To provide the student with familiarity in the use of standard notations and models for the detailed design of software;

b) To provide students with experience in using (automated) techniques in the implementation of rigorously specified software.

c) To introduce students to C in order to utilise the widely available implementation tools based on C.

d) To acquaint student with the problems of documenting and maintaining large software systems.

Syllabus:

Part A

1. Program Design

- algorithmic notations: extended grammars, decision tables, PDL, data encapsulation
- interaction Handling: the Finite State Machine, input analysis (via e.g. awk, lex),screen handling and window management (e.g. FORMS, Gem, X windows).
- exception handling
- concurrency - CSP and the monitor concept.

2. Implementation

- code generation: language issues, compiler characteristics, structure editors,code generation tools (e.g. KINDRA)
- Unit testing techniques: use of debuggers
- integration: configuration management,integration testing, performance monitoring, source code control (e.g. SCCS).

3. Project Management

- goals and milestones.
- documentation requirements.
- quality assurance: test and maintenance planning, acceptance testing.
- introduction to working within an IPSE - e.g. Genos.

Part B

4. Programming in C as a conversion course

- using the C preprocessor.
- building modular systems.
- object oriented programming.
- interfacing between the language, operating system and the command processor.
- using the Programmer's Workbench.

Indicative Coursework

Starting with a medium sized application system, specify and implement several different user interfaces (e.g. command driven, menu driven). Port the system onto another machine and provide documentation for configuration control.

As a class, generate a project plan for the detailed design, implementation and testing of a piece of software required as a result of an analysis carried out in Unit 3.
Individually implement, test and document component modules. As small groups, integrate and document these components.

Indicative Booklist

R. Fairley "Software Engineering Concepts" McGraw Hill 1985

R.S. Pressman "Software Engineering: a Practitioners Approach" McGraw Hill 1982

N. Gehani, D. McGettrick (editors) "Software Specification Techniques" Addison Wesley 1986

J.G. Janossy "Commercial Software Engineering" Wiley 1985

S.R. Bourne "The Unix System" Addison Wesley 1983

B. Kernighan, D.M.Ritchie "The C Programming Language" Prentice Hall 1978

I.C. Pyle "The Ada Programming Language"
Prentice-Hall 1981

Class Contact

42 hours = 28 hours lecture + 14 hours tutorial

Assessment

Examination - 50%
Coursework - 50%

Unit 5: OPTIONS

Postgraduate diploma options are intended to be intensive courses covering some specialised and fairly advanced topic. In order to keep the course fresh and up-to-date, these options will be put forward from time to time by contributing members of staff and validated by the Polytechnic Academic Standards Committee.
This will enable staff to offer courses in their specialisms or in new areas where they are not yet ready to mount a more formally defined course. Alternatively, this scheme will enable the course team to construct options which from year to year have direct relevance to the goals and aspirations of a particular class.

Format for Mini-Option Proposal

In order to help the students assess the merits of different options submitted by different members of staff, the description of an option proposal should be more explicit than the conventional syllabus. The format for an option submission is fixed as follows:

Option Title: .....

Synopsis

A brief description of

(i) how the course will be conducted in terms of lectures/tutorials/seminars/practicals;
(ii) the material to be covered;
(iii) where the subject matter fits into Computing as an academic discipline, the job market, or any other relevant reference points.

Course Breakdown

A detailed account of the way the course will run - the order in which topics will be covered,preferably on a lecture-by-lecture or at least a week-by-week basis.

Introductory reading

A single reference which introduces the subject at an appropriate level. The idea here is that the student faced with a difficult choice of options can investigate the subjects in a little bit more depth in advance. The introductory reference should be about an evening's reading. (It could be a set of photocopied articles or the first chapter or so of a good textbook.)

Reading List

Normal reading list - with any recommended textbook indicated.

Typical Assignment

Typical Exam. Question

Unit 5 Currently Validated Options

Option Title: Computer Graphics

Synopsis

The course covers the elements of computer graphics, concentrating on those features found in most two-dimensional, device-independent graphics libraries. Some attention will also be given to techniques for graphics program design. Practical sessions will allow students to familiarise themselves with different graphics input/output devices. Computer graphics is an area of computing which has gained importance with the lowering prices of display hardware and high-performance workstations. Significant application areas include CAD and desktop publishing.

Course Breakdown

1: Introduction - Display Devices; Achieving device independence.
2: Basic Graphics - Output primitives and attributes.
3: The Viewing Operation - Internal representation of graphical information.
4: Two-dimensional coordinate transformations.
5: Graphical Input - Devices.
6: Graphical Input - Techniques.
7: More Advanced Features - Viewports and Windows; Clipping.
8: More Advanced Features - Segmentation.
9: More Advanced Features - three-dimensional viewing operations.
10: Program Design for Graphics Output - designing the display.
11: Program Design for Graphics Input - designing the dialogue.
12: Graphics Library Packages.

Introductory Reading

Chapter 1 of "Computer Graphics" by D. Hearn and M.P.Baker, Prentice Hall International 1986.

Reading List

D. Hearn and M.P.Baker	"Computer Graphics" Prentice Hall International 1986
J.D.Foley and A.van Dam	"Fundamentals of Interactive Computer Graphics" Addison-Wesley, 1982.
R. Salmon and M. Slater	"Computer Graphics: Systems and Concepts" Addison Wesley 1987.

Typical Assignment

Write a program which illustrates some graphical technique (e.g. clipping or dragging) or some feature (e.g. zooming and panning) associated with a specific graphics display device.

Typical Exam. Question

Describe the five logical classes of graphical input device. State the type of input which each is designed to accept and name a physical device which corresponds to each class. For each logical class, describe a method whereby the input operations appropriate to that class may be simulated by a physical device belonging to another class. Explain how the concept of logical input classes contributes to the design of interactive graphics packages.

Option Title: Functional Programming

Synopsis

Functional programming offers a new look at computing. The major goals of functional programming are to provide structurally transparent languages so programs can be verified and optimised mechanistically and to facilitate the implementation of parallelism in the coming generation of computers. The way that functional languages attempt to achieve these goals is by separating the task that the program is to perform from the way the computer is to do it. That is, unlike imperative languages (Pascal, ADA, Modula2 etc.,) functional languages do not specify the flow of control, but only the flow of data within a program. Current well known functional languages include: LISP, ML, HOPE, MIRANDA and FP.

Course Breakdown

1: Introduction to HOPE and recursion
2: Simple recursive functions on the natural numbers
3: Type polymorphism
4: Pattern matching
5: Recursive data structures - lists, trees etc.
6: Lambda notation
7: Higher order functions
8: Correctness of functional programs (induction proofs)
9: Lazy evaluation
10: Functional language interpreters
11: A survey of different functional languages
12: Fifth generation architectures

Introductory Reading

S. Eisenbach and C. Sadler — "Declarative Programming" BYTE Vol 10 No. 8 August 1985

Reading List

Glaser, Hankin and Till — "Principles of Functional Programming" Prentice Hall
P. Henderson — "Functional Programming - Application and Implementation" Prentice Hall
M.C.Henson — "Elements of Functional Languages" Blackwell

Typical Assignment

Write a program to check whether or not a word is a palindrome. Prove by induction that your program is correct in the sense that palindrome(1) <=> palindrome (reverse(1))

Typical Exam. Question

What are the major differences between functional and imperative programming languages? Explain why functional languages may provide a solution to the software crisis.

Option Title: Data Modelling and Further Database Theory

Synopsis

This option covers the theory of data modelling and database systems. It builds on, and extends the database work done in Unit 3. It covers the theory of data models from a formal perspective in some detail, discussing structures, constraints and operations as found in various data models including relational models, network models, data models which use subtyping and others such as the binary data model and the functional data model. The implications and consequences of referential integrity constraints in the relational data model are considered in depth.

Course Breakdown:

1. Formal Theory of Data Models as consisting of structures, constraints and operations.
2. The relational approach to database systems; its structures, constraints and operations.
3. The network approach to database systems; its structures, constraints and operations.
4. Data models which include subtyping; Codd's RM/T model; its structures, constraints and operations.
5. Detailed discussions of the implications and consequences of referential integrity constraints in relational data models; Cascading; The Industry Standard SQL2.
6. Other approaches such as the binary data model, the functional data model; their structures, constraints and operations.

Introductory Reading

An Introduction to Database Systems by Chris Date (vol 1) Fourth Edition. Addison Wesley 1986

Reading List

Chris Date — "An Introduction to Database Systems" Fourth Edition. Addison Wesley 1986

D.C.Tsichritzis and F.H.Lochovsky — "Data Models" Prentice Hall 1982

Peter Gray — "Logic, Algebra and Databases" Ellis Horwood 1986

Typical Assignment

Given a simple data model, to implement this using the network and relational database systems and thereby to compare the two systems

Typical Exam. Question

Discuss and compare the relational and network data models with respect to their structures, constraints and operations.

Explain the term referential integrity and discuss its implications with respect to cascading. Discuss how the industry standard SQL2 handles referential integrity constraints.

Unit 6: Project

Aims:

To give students an opportunity to integrate the knowledge and skills gained in other units and to exercise these in a realistic situation.

To provide students with the experience of undertaking a complete and sustained piece of work, from planning through realisation to report-writing.

Implementation

Students will select a project from a list of titles supplied by lecturers teaching on Units 3,4 and 5 (the option). Normally individual projects will be undertaken, although in special circumstances, a group project may be proposed.

Supervision and assistance will be provided up until the end of the summer term when all the practical work will normally be expected to have been completed. Students will submit project documentation not later than the beginning of the next session.

Class Contact:

45 hours = 15 hours project supervision
+ 30 hours practical project work

plus 30 hours write-up time in the summer

Assessment:

All students will have a viva during the last week of term when the practical work of the project will normally be expected to have been completed, but before the summer 'write-up' period. Students will be expected to provide a verbal report, where appropriate to present provisional 'results', and to give some idea of the format of the written report. During the viva, students may be advised about how the write-up should be approached or about any additional practical work which the panel may feel necessary to complete the project.

Viva - 20%
Report - 80%

PROVISION OF WRITTEN EVIDENCE TO EXAMINATION BOARDS

Examination Boards are able to take account of special circumstances, such as illness or domestic problems, which might have affected a student's performance. However, it is YOUR responsibility to inform your Course Tutor or Head of Department of any such circumstances, as soon as possible after the examination or the submission of coursework.

If you think your performance has been affected by illness, personal difficulties, or other such cause, you should explain this in a letter, and give this to your Course Tutor or Head of Department immediately after the examination. Your Course Tutor/Head of Department will make this available to the Examination Board. The Examination Board will give careful consideration to all information about special circumstances. This does not mean that the Board will automatically excuse a failure because a student has had particular problems. But the Board obviously cannot make any allowance for special circumstances if it knows nothing about them.

Information on special circumstances MUST be notified to your Course Tutor or Head of Department before the meeting of the Examination Board. The Examination Board will only consider special circumstances which have been notified after its meeting if there is good reason for the information not having been made available earlier. "Good reason" does not include ignorance of the Regulations. You will also not be able to ask an Examination Board to reconsider its decision if your grade is lower than expected, on the grounds that your performance was affected by special circumstances: it is your responsibility to bring such information to the attention of your Course Tutor/Head of Department before the Examination Board meets.

Many students, I know, feel embarrassed about submitting extenuating circumstances. But it is always to your advantage to do so if your work has really been affected by illness or some other such cause. You must not assume that your personal problems are too trivial or too personal to put to an Examination Board. If you are in any doubt about your situation, you should ask your personal tutor or course tutor, or the Student Counselling and Advisory Service.

If you are ill, even if only for one day during an examination, you should go to your doctor and obtain a certificate to submit to the Board. You should do this whether you think you will pass or fail - if you wait until you discover you have failed, you will be too late. If you are ill at any other time in the course, and feel that this illness may have affected your performance, you should also report this and provide a certificate.

Any evidence provided by a student will be treated by the Examination Board as strictly confidential, and will in no circumstances be made known to staff who are not members of the Board, or to fellow students.

Peter Holliday
Registrar

10.6.87

[8a:extencir.mem]