INFORMATICS FOR SECONDARY EDUCATION

A Curriculum for Schools

Produced by a working party of the
International Federation for Information Processing (IFIP)
under the auspices of UNESCO

Working Party Members:

Ulrich Bosler, Germany
Sam Gumbo, Zimbabwe
Harriet Taylor, USA
Zoraini Wati Abas, Malaysia

Charles Duchâteau, Belgium
Raymond Morel, Switzerland
Peter Waker, South Africa

Chair & Joint Editor: Tom van Weert, Netherlands
Joint Editor: David Tinsley, United Kingdom

The Chair of the Working Party wishes to thank the Centre Informatique Pédagogique, Geneva, Switzerland for its support during the preparation of this document

Paris 1994

Intergovernmental Informatics Programme

ED/HEP

Division of Higher Education

International Federation for Information Processing (IFIP)

ED/94/WS/ *12*

The opinions expressed in this document are those of the authors and do not necessarily reflect the views of the UNESCO Secretariat.

INFORMATICS FOR SECONDARY EDUCATION

A Curriculum for Schools

Contents

Appendices

INFORMATICS FOR SECONDARY EDUCATION

A Curriculum for Schools

Introduction

General Aim

UNESCO aims to ensure that all countries, both developed and developing, have access to the best educational facilities necessary to prepare young people to play a full role in modern society and to contribute to wealth creation.

Information technology (IT) has become, within a very short time, one of the basic building blocks of modern industrial society. Understanding IT and mastering the basic skills and concepts of IT are now regarded by many countries as part of the core of education alongside reading and writing. This area of study goes under the all-embracing name of informatics.

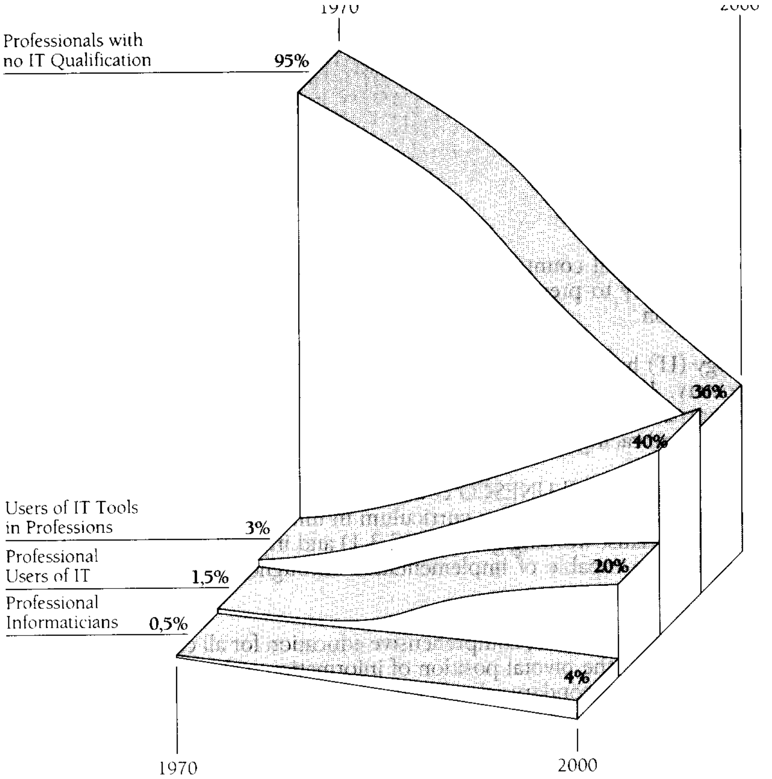
To give positive, practical help to all UNESCO countries, the International Federation for Information Processing (IFIP) has been asked to specify a curriculum in informatics for secondary education based on the experience of its specialist working group (WG 3.1) and input from other selected experts. The curriculum is designed to be capable of implementation throughout the world to all secondary age students.

All governments aim to provide the most comprehensive education for all citizens within the constraints of available finance. Because of the pivotal position of information technology in modern societies, the introduction of informatics into secondary schools will be high on any political agenda. This document gives a practical and realistic approach to the informatics curriculum which can be adopted quickly and at minimum cost.

Justification for Informatics Learning

Information technology now permeates the business environment and underpins the success of modern corporations as well as providing government with cost efficient civil service systems. At the same time the tools and techniques of information technology are of value in the processes of learning and in the organisation and management of learning institutions.

The trends in the need for qualified personnel give a clear demonstration of the need for effective informatics learning at all levels (source: Schul Computer Jahrbuch, Ausgabe '93/94, Metzler Schulbuch Verlag, page 15):



professional informaticians (computer scientists)

1970: 0,5 % of the professional population
2000: 4 % of the professional population

professionals in other disciplines with an informatics qualification

1970: 1.5 % of the professional population
2000: 20 % of the professional population

professionals competent in the use of informatics tools

1970: 3 % of the professional population
2000: 40 % of the professional population

professionals with no IT qualification

1970: 95 % of the professional population
2000: 36 % of the professional population

In light of these clear indications it is timely for UNESCO to promote the introduction of informatics in all secondary schools in all countries. The commissioning of IFIP Working Group 3.1 to design the recommended curriculum draws on their wide experience of developing effective informatics education during the past quarter of a century.

Tom van Weert Chair, IFIP Working Group 3.1 on Secondary Education

Section 1—The Curriculum Format

Aim

The working party has aimed to produce a document which defines a practical and realistic approach to the secondary education informatics curriculum for both developed and developing countries which can be implemented, according to available resources, quickly and at minimum cost.

The curriculum has been designed in modular form so that education authorities can select appropriate elements to meet their objectives at the phase of development reached in their countries. Sufficient detailed description of each objective has been given so that writers can produce course materials which meet local cultural and developmental circumstances.

In any educational system, the level of available resources places a restriction on the degree to which any new subject can be introduced into the school curriculum, especially where only the most basic facilities have so far been provided. But informatics is of such importance to the future industrial and commercial health of a country that investment in the equipment teacher training and support services necessary for the effective delivery of an informatics curriculum should rank high in any set of government priorities. The curriculum proposed has taken account of these resource issues and has specified a minimum requirement for effective delivery in different circumstances.

Background Considerations

Introducing any new curriculum calls for careful preparation, management, resourcing, teacher training and continuing support. Experience shows that informatics is no exception. **Many of its elements may find their most effective place within other, existing subject areas. The proposed informatics curriculum could, for example, to a large extent be embedded within mathematics, science, languages and social science**. Where this is not practical, or timely with respect to the phase of development (see below), it is recommended that the curriculum be delivered as a separate entity, calling where possible on other subjects for practical illustration and example.

Different countries will be at different phases of development in the use of information technology within government, commerce and wider society. Three distinct phases have been identified:

Automation Phase—where essential infrastructures are still being developed and conversion of existing systems and design of informatics solutions is the sole responsibility of technical personnel.

Information Phase—characterised by the move towards personal ownership or use of computing tools with a strong influence of the user on the design of automated systems.

Communication Phase— the most advanced stage when computers are in networks and use is characterised by collaboration between users and informatics is part of the essential infrastructure.

To help support implementation in particular countries, the curriculum has been specified for countries in the Information Phase. Adaptation will be necessary for countries in the earlier or later stages of development. Further advice on matters of implementation is given in Section 4.

Design of the Informatics Curriculum

The recommended curriculum has been arranged in the form of Units, grouped together in Modules which are designed for different levels of secondary education. Depending on local circumstances, however, the units could be used at other levels.

Foundation Level Modules are for all students up to the age when a choice is made either to stay in full time schooling or to seek work (usually at about the age of 16). Units for the foundation stage have been further grouped into a Computer Literacy Core Module, which is regarded as General Education for all students, together with a Computer Literacy Core Elective Module which is also targeted at General Foundation Level. A commentary is provided which indicates which of the above modules are of a Vocational nature, appropriate for students who will leave school for work after the Foundation Level.

Optional Modules of Computer Literacy are targeted at either Foundation or Advanced Levels; also two Units on Programming which are pre-requisites for those proceeding to Advanced Level Informatics.

Advanced Modules are for secondary school students who stay in full time education. Here again, there is a grouping of Units into a General Module aimed at students aspiring to higher education and a Vocational Module for students entering the work place after a period of technical training.

Terminology

The following definitions have been used by the working party:

Informatics: the science dealing with the design, realisation, evaluation, use and maintenance of information processing systems; including hardware, software organisational and human aspects, and the industrial, commercial, governmental and political implications (UNESCO/IBI).

Informatics technology: the applications of informatics in society.

Information technology: the combination of informatics technology with other, related technologies.

Section 2—Main Objectives of the Informatics Curriculum

1. COMPUTER LITERACY

Students should be able to use computers in a competent and intelligent way in daily life.

Recent radical changes in the work place and in the qualifications required for effective performance mean that **secondary schools should aim to include at least the core elements of computer literacy as defined here within their main core curriculum.**

This main objective should preferably be addressed within General Education at the Foundation Level.

2. APPLICATION OFIT TOOLS IN OTHER SUBJECT AREAS

Students should be able to use information technology tools to solve routine problems in other subject areas.

This main objective should preferably be addressed within General Education at both the Foundation and Advanced Levels.

3. APPLICATION OF INFORMATICS IN OTHER SUBJECT AREAS

Students should be able to use methods and techniques from informatics in combination with information technology tools to solve problems in other subject areas.

This main objective should preferably be addressed in General Education at the Advanced Level

4. APPLICATION OF INFORMATICS IN PROFESSIONAL AREAS

Students should be able to use methods and techniques from informatics in combination with information technology tools to solve professional problems from business and industry.

This main objective should preferably be addressed in Vocational Education at the Advanced Level

Curriculum Objectives within the Secondary Education Context

GENERAL EDUCATION AT THE FOUNDATION LEVEL

The principal focus is on Main Objectives 1 and 2:

Computer Literacy

Application of IT tools in other subject areas

In addition, there is an optional focus on Objective 3:

Application of Informatics in other subject areas

GENERAL EDUCATION AT THE ADVANCED LEVEL

Having fully met Objective l (Computer Literacy), the focus is on Main Objectives 2 a 3:

Application of IT tools in other subject areas

Application of Informatics in other subject areas

Objective 3 involves the following sequence of problem solving skills using the techniques and tools of informatics, namely the methodical modelling of the problem, design of an algorithmic solution, programming the solution either in a general or computer specific way, and an evaluation of the proposed solution. This implies that students have developed a functional model of a computer system and its programming environment.

VOCATIONAL EDUCATION AT THE ADVANCED LEVEL

Here, the focus is on Main Objective 4:

Application of Informatics in professional areas

Students should be able methodically to model, design, realise and implement relatively simple information systems with the aid of problem oriented tools; and can identify problems involved in project management.

Section 3—The Curriculum Units

GENERAL EDUCATIONAT THE FOUNDATION LEVEL

Computer Literacy

Objectives

Students should be able to:
1. handle the basic hardware and software facilities of a computer system;
2. use, control and apply application oriented software tools
3. solve routine problems in an algorithmic form;
4. identify the most important social, economical and ethical consequences of IT.

Units of Computer Literacy for the Foundation Level have been divided into two modules:

CORE MODULE

Cl* Hardware
C2* Systems Software Environment
C3* Computing Trends
C4 Introduction to Using a Computer
C5 Text Processing
C6 Working with a Database
C7 Working with Graphics
C8 Social and Ethical Issues
C10 Choice of Software Tools

CORE ELECTIVE MODULE

E1 Database Design and Use
E2 Spreadsheet Design awl Use
E3 Careers in Informatics

*** These Units could be integrated with other Core Units as and when applicable.**

In addition, two Optional Modules are specified for use either at the Foundation Level or the Advanced Level:

OPTIONAL PROGRAMMING MODULE (SEE COMMENTARY ON IMPLEMENTATION BELOW)

P1 Introduction to Programming
P2 Top-Down Program Design

GENERAL OPTIONS MODULE

Opl Keyboarding Skills
Op2 Desk Top Publishing
Op3 Computers and Communication
Op4 Creating Graphics
Op5 Working with Multimedia
Op6 Computer Aided Design (CAD)
Op7 Modelling and Simulation
Op8 Expert Systems
Op9 Robots and Feedback Devices
OplO Music
Op11 Statistics

FULL SPECIFICATIONS OFALLTHE ABOVE UNITS ARE GIVEN IN APPENDIX 1

Implementation of Foundation Units :

Recent radical changes in the work place and in the qualifications required for effective performance mean **that secondary schools should aim to include at least the core elements of computer literacy as defined here within their main core curriculum.**

This means that all schools should aim to cover the Core Module at a Foundation Level for all students. The Core Elective Module gives a broader experience of database and spreadsheet design and use, of particular value to those seeking employment as data entry clerks or for future secretaries; as well as a module on careers in informatics.

Units for Vocational Courses

The Core Elective Units will be of special interest to students leaving for work at the end of the Foundation Level. The following Optional Units will also be of value in vocational courses:

Keyboarding Skills (Opt) combined with the Core Unit of Text Processing (C5) may lead to employment as a secretary or data entry clerk, and will help gain entry into vocational training institutions which provide further training for secretaries.

Desk Top Publishing (Op2) and Creating Graphics (Op4) combined with the Core Unit on Working with Graphics (C7) will be of advantage to those seeking employment in the graphic design and printing industries.

Working with Multimedia (Op5) will help students to look for work in the advertising industry or the public relations department of a large firm.

Computer Assisted Design (Op6) will be of immediate value to those seeking employment in engineering, design, architecture and other occupation requiring the creation of technical drawings.

Informatics in Other Disciplines

Computer Literacy Objective 2 and Main Objectives 2 & 3 require students to apply IT Tools and Informatics Techniques within other subject areas. At a Foundation Level, there requirements can be met by using a selection of the examples given in Appendix 2—Informatics in other Disciplines. Students will also find these examples a stimulant to their own work in other subjects as well as an enrichment of their informatics studies.

Teachers may find that by integrating the use of computers within subject areas, **most of the Computer Literacy objectives can be met without the need for a separate course.**

Optional Programming Module

Computer Literacy Objective 3 requires students to have the ability to solve routine everyday problems in an algorithmic form. This is most easily achieved by including the two Units P1 and P2 of the Optional Programming Module within the Foundation Level Core Module. In this context, the word algorithm is to be interpreted in its broadest sense, avoiding a narrow mathematical definition. Depending on the local situation and the availability of resources, this objective can be met either by the Programming Module, or by stressing the problem solving nature of using software tools within the main Core Units, thus avoiding too formal a treatment of algorithms at this stage. As an alternative, this objective could be met within the Core Mathematics Curriculum.

General Options Module

Is is recommended that these Computer Literacy Units be used at either the Foundation or the Advanced Levels, subject to the specific requirements at Foundation Level described above.

All Units should be studied by those wishing to have a broad understanding of computers in modern life but it is recognised that the resources available may limit the degree to which they can be supported by practical exercises within secondary schools.

Resource Requirements

In Appendices, each Unit Description indicates the minimum necessary resources required for successful implementation, and gives suggestions for optional extra resources.

ADVANCED LEVEL UNITS

For completeness, Advanced Level Units are specified here to help older students to bridge from foundation level courses to general and vocational courses at the tertiary stage. It is recognised that not all secondary schools will be staffed and equipped to provide advanced units.

Ideally, courses should be built up from General Advanced Units GA1-3 and Vocational Advanced Units VA1-3, in consultation with universities and tertiary institutions, so that advanced credits can be obtained towards a tertiary computer science course.

GENERAL EDUCATION AT THE ADVANCE LEVEL

The **Computer Literacy** objective should have been met at the Foundation Level.

Application of IT tools in other subject areas

Students should be able to use information technology tools to solve problems in other subject areas.

Application of Informatics in other subject areas (selected students only)

Students should be able methodically to model and solve relatively complex problems using both general purpose and problem oriented programmable tools.

COMPUTER LITERACY MODULES

Both Units specified in the Optional Programming Module (Pi and P2) should either have been studied at Foundation Level or must be studied first at this level. Also, subject to the availability of resources, all Units from the General Options Module (Opl-11) should be studied to meet the Computer Literacy objectives at this level.

GENERAL ADVANCED MODULE

Three Units are specified to meet the objectives for General Education in Informatics at the Advanced Level:

GA1	Foundations of Programming and Software Development
GA2	Advanced Elements of Programming
GA3	Applications of Modelling

FULL SPECIFICATIONS OFALLTHE ABOVE UNITS ARE GIVEN IN APPENDIX 3

Application of IT Tools in other subject areas

To meet this objective, students should have broad experience in the use of IT tools in other disciplines as described in Appendix 2; and should have studied Unit GA3, using Subject Oriented Programming (see Unit description).

Application of Informatics in other subject areas

For a fuller treatment of Informatics, students should complete all three Units GAl, GA2 and GA3. This will enable students to enter Higher Education with basic knowledge and skills in the programming of systems and in software development, as well as with practical experience in modelling.

Implementation

Both of the Units (Pi and P2) from the Optional Programming Module should be studied before attempting these General Advanced Level Units in Informatics.

Unit GA3: Applications of Modelling can be studied in parallel with Unit GA2 to give a more realistic context to the work on Advanced Elements of Programming.

VOCATIONAL EDUCATION AT THE ADVANCED LEVEL

Objective

Students are able methodically to model, design, realise and implement relatively simple information systems with the aid of problem oriented tools; and can identify problems involved in project management.

Three Units are specified to meet the objective for Vocational Education in Informatics at the Advanced Level:

VOCATIONAL ADVANCED MODULE

VAl Business Information Systems
VA2 Process Control Systems
VA3 Project Management

FULL SPECIFICATIONS OFALLTHE ABOVE UNITS ARE GIVEN IN APPENDIX 4

Implementation

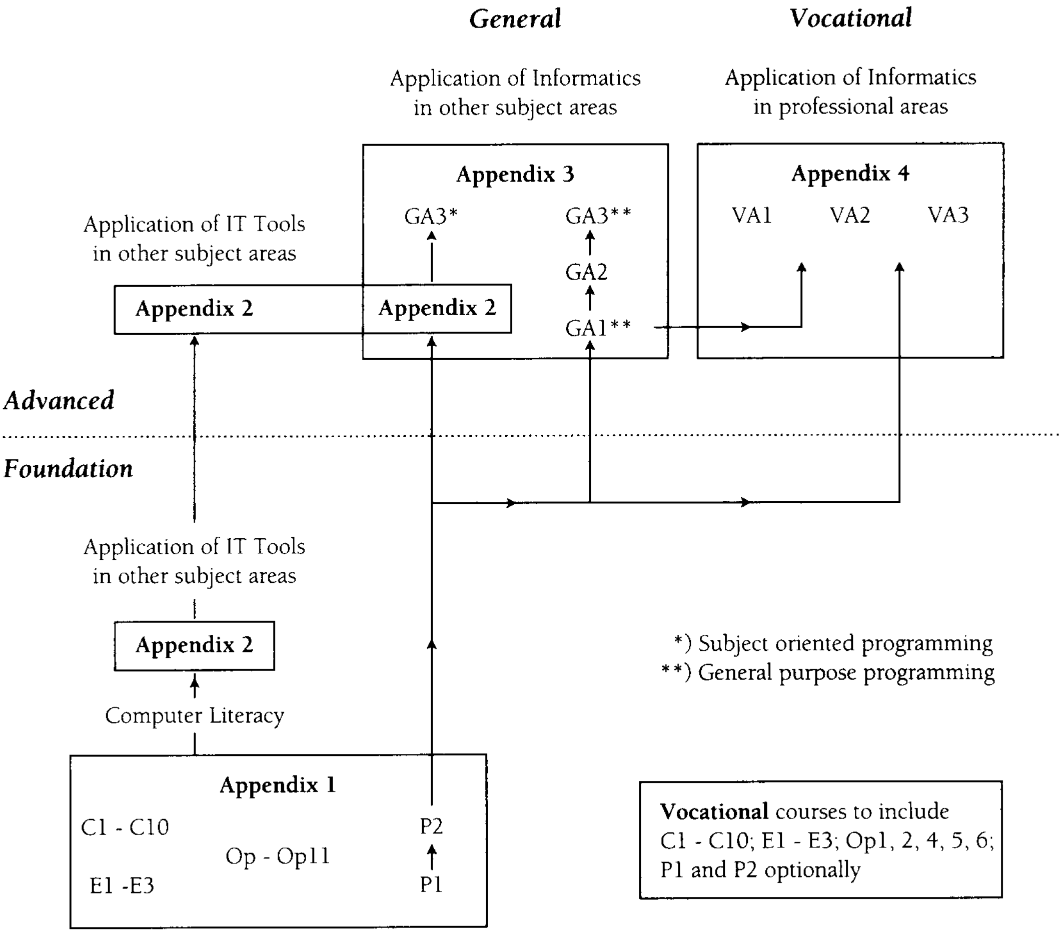
Both of the Units (Pl and P2) from the Optional Programming Module should be studied before attempting this Vocational Advanced Level Module in Informatics.

The Units recommended as optional for study by vocationally oriented students at the Foundation Level should now be regarded as pre-requisites.

A higher technical level of competence can be achieved by studying Unit GAl from the General Advanced Level Module before starting on VAl, VA2 and VA3.

Instead of studying Units VAl, VA2 and VA3 in sequence, Unit VA3 may be used to provide a realistic context within which the objectives of Units VAl and VA2 can be met.

Relationships between the Units


General
Vocational
Application of Informatics in other subject areas
Application of Informatics in professional areas
Appendix 3
GA3*
GA3**
GA2
GA1**
Appendix 4
VA1
VA2
VA3
Application of IT Tools in other subject areas
Appendix 2
Appendix 2
Advanced
Foundation
Application of IT Tools in other subject areas
Appendix 2
Computer Literacy
Appendix 1
C1 - C10
Op - Op11
E1 -E3
P2
P1

*) Subject oriented programming
**) General purpose programming

Vocational courses to include C1 - C10; E1 - E3; Op1, 2, 4, 5, 6; P1 and P2 optionally

Section 4—General Implementation Issues

It is recognised that all developing countries will wish to adopt a curriculum which helps them rapidly to catch up with more technologically advanced countries. But it is important that the early steps in the developmental process are not ignored entirely, otherwise a dependency on other countries for technical support could be an unwelcome outcome.

Schools should be aware that developments connected with informatics in education can be very fast, and may also be separated in phases similar to the phases of information technology development in wider society. The following descriptions are provided to help determine in which phase of development countries have reached, so that they can decide on appropriate action.. The transition between phases is often gradual, but if schools and authorities are aware of which direction developments will take, they will be able to plan future actions.

Automation Phase

Only a few computers are available for a large number of students, often not of the latest generation of computer, with slow and low quality printing facilities. Application software is based on general purpose programming languages (Basic, Pascal), typically a simple word-processor, a simple file system, possibly a simple spreadsheet. Teachers have limited opportunity for training and external support, often schools have only one specialist teacher. Computers are either all located in a specialist room or dispersed as one per classroom. Students have no prior exposure to computers and have little or no keyboard skills. Informatics is usually a separate course with limited integration in other subjects.

In curriculum terms, this phase calls for units on computer programming for selected students and consideration of the social effects of introducing automated systems. Resources available for units on Computer Literacy may not be available.

IFIP Guidelines on Good Practice Integrating Information Technology into Education (see Appendix 5) gives practical advice on planning for the transition from the automation phase to the next, information phase.

Information Phase

Schools have become major users of personal computers, and most subjects have started to integrate information technology elements either for learning support or as curriculum content. National and regional policies are in place for the evaluation and supply of hardware and software, and information about systems and applications circulates freely.

The main characteristic of the Information Phase in education is the interaction between the organisation of education and information technology. In the Information Phase education is moving from teacher centred towards student centred education under the pressure of society which needs students with other competences, this movement is supported by information technology. On the other hand the increase in technical capabilities and in the use of information technology pushes educational organization in the same direction. The Information Phase is characterised by the integration of information technology into education.

Schools will have sizable numbers of computers available for their students. These computers will be accessible to students on an individual basis, and be dispersed through the school. Computers will be tied into a local area network with filing and printing facilities. A whole range of advanced application software will be available to the students. School staff, teachers and management, use computers for support in their work. Specialist maintenance support will be available. There are several teachers in the school who are specialists in computer use in the teaching of their discipline. Students may well have already acquired computer skills, either in education or at home. Informatics is, for an important part, integrated into other disciplines.

The secondary curriculum now includes general computer literacy for the majority of students, and the use of application tools within subject disciplines as well as methods and techniques from informatics such as programming.

More about the problems and possible solutions associated with the Information Phase may be found in the IFIP Guidelines for Good Practice Integration of Information Technology into Education — see Appendix 5.

Communication Phase

The Communication Phase is at the moment emerging in isolated experiments and developments. Precisely how education will be affected is not yet clear. However, this phase will be characterised by full integration of information technology in the management and delivery of learning, and in the use of networks of computers for communication and collaboration between computer users, both within the school and with outside agencies.

In schools, teachers and students will be linked in computer networks. The secondary curriculum will include general Information Technology Literacy, which is broader and deeper than Computer Literacy. Information Technology will be fully integrated in the learning process. Use of Information Technology will be fully integrated in the different subject areas, and advanced creative use' will be made of these tools on the basis of modelling methods and techniques from informatics. Application Oriented Informatics will be an elective.

Evolution of Information Technology and Education

Requirement	Automation Phase	Information Phase	Communication Phase
To know about ...	theory of computer architecture; formal processing; history of informatics and information technology	what tool for what kind of task ? (Some operations, with their characteristics, are hidden under the user interface; we are evolving from an analogic world into a digital one: clock, phone, CD ...)	network architecture
To do with ...	text editor, compiler; classical programming environment	different interfaces (languages); tools (WP, spreadsheet ...) used creatively, rationally	information retrieval; collaboration

To have it done by ...	algorithms;
	procedural programming;
	Logo;
	robotics
		macros (programming by "recording");
		programming through word-processing, spreadsheet ...
			object oriented processes

Key features	No personal use;
	Tasks have to be programmed in avance... not managed by dialogue
		Single user;
		Tasks done by dialogue;
		Use of tools
			Integration;
			Resource sharing;
			End of individual computer

Aims :

1. To prepare for the future without dreaming about the present
2. To detect and interpret non-changing features

Further Practical Considerations

Informatics is essentially a practical subject. Informatics skills are best acquired through practical work on and with computers; informatics knowledge develops more effectively within a practical environment.

This poses problems when there is a limit to the supply of equipment to support informatics courses, but it is important to have an implementation strategy which acknowledges the need for practical experience and arranges for the supply, management and maintenance of computing equipment.

If equipment is in limited supply, at least each teacher of informatics should have ready access to computing equipment, preferably in advance of its introduction to students. Ideally this should be associated with a formal course of training so that teachers are thoroughly familiar with the content and practice of informatics. Many inspired teachers will then find ways to compensated for lack of equipment for students until the supply improves. It must be stressed that an adequate supply of equipment is a necessary condition for schools wishing to enter the Information Phase.

The Curriculum Units indicate examples of applications which help to place the objectives in context. Not all countries will be capable of implementing all of the suggestions, but teachers will be able to find examples of a similar sort which match more closely the culture and experience of their own students.

Resource Centres

Several countries, where informatics has been successfully introduced, have established resource centres for the initial training of teachers, the acquisition, evaluation and dissemination of hardware and software, and the preparation, printing and distribution of resource materials for teachers and students, ideally involving teachers in the development and evaluation phases.

Resource centres can be national or local. They represent good value for money at the beginning of the introduction of an informatics curriculum, and can be an important continuing source of support for teachers as technology changes and the country moves towards the communication phase. Good examples also exist where the facility to communicate via electronic mail between the teacher's computer and the resource centre has helped to update local skills and knowledge without the costs of travel and time away from school.

In spite of any difficulties which teachers may find when planning to introduce the recommended curriculum, they can take encouragement from the enthusiasm which most students bring to their learning of informatics. The subject is contemporary, closely linked to the real world of today and tomorrow, and its study will ensure that students are equipped to play a key role with confidence in their future society. Teachers will enjoy the challenge of learning informatics for themselves and will soon capitalise on the motivation of their students.

Implementation Stage —>	**Initial**	**Second**	**Third**	**Advanced**

	Insufficient hardware;
	Few suitable teachers;
	Some enthusiasts; No Computer Science Graduates
		More hardware;
		In-service trained teachers (one per school);
		No Comp. Science Graduates
			More hardware;
			More in-service trained teachers;
			Few Comp. Science Graduates
				More advanced hardware;
				Most teachers computer literate;
				More Computer Science Graduates

Junior Secondary Ages 12 - 16		**All students** CORE Module Units C1 - C10	**All students** CORE Module Units C1 - C10 **All students** ELECTIVE Module Units E1 - E3	**All students** CORE Module Units C1 - C10 **All students** ELECTIVE Module Units E1 - E3 **Selected students** OPTIONAL PROGRAMMING Units P1 - P2
Senior Secondary Ages 16 - 19	**Most students** CORE Module Units C1 - C10 **Selected students only** ELECTIVE Module Units E1 - E3	**All students** ELECTIVE Module Units E1 - E3 **Most students** OPTIONS Module Units Op1 - Op11 **Selected students** OPTIONAL PROGRAMMING Units P1 - P2	**All students** OPTIONS Module Units Op1 - Op11 **Selected students** OPTIONAL PROGRAMMING Units P1 - P2	**Most students** OPTIONS Module Unit Op1 - Op11 **Selected students** ADVANCED Modules Units GA1 - GA3 Units VA1 - VA3 *Leading to Advanced Credits*
	All programming taught at tertiary level	Students enter tertiary level with some programming experience	Students enter tertiary level with some programming experience	Some students might obtain Advanced Credits for tertiary courses while still at school

An illustration of how Units might be introduced over time as resources and facilities become available

Introduction
Curriculum Format
Main Objectives
Curriculum Units
General Implementation Issues
Appendices

Appendices

Appendix 1—Computer Literacy Units

Contents

CORE MODULE

C1* Hardware
C2* Systems Software Environment
C3* Computing Trends
C4 Introduction to Using a Computer
C5 Text Processing
C6 Working with a Database
C7 Working with a Spreadsheet
C8 Working with Graphics
C9 Social and Ethical Issues
C10 Choice of Software Tools

CORE ELECTIVE MODULE

E1 Database Design and Use
E2 Spreadsheet Design and Use
E3 Careers in Informatics

* These Units could be integrate with the other Core Units as applicable.

GENERAL OPTIONS MODULE

Opl Keyboarding Skills
Op2 Desk Top Publishing
Op3 Computers and Communication
Op4 Creating Graphics
Op5 Working with Multimedia
Op6 Computer Aided Design (CAD)
Op7 Modelling and Simulation
Op8 Expert Systems
Op9 Robots and Feedback Devices
Op 10 Music
Op 11 Statistics

OPTIONAL PROGRAMMING MODULE

Pl Introduction to Programming
P2 Top-Down Program Design

Appendix 2 gives examples of how the above Computer Literacy Units can stimulate the use of computers in other disciplines.

Units forAdvanced Level Courses are specified in Appendices 3 and 4:

GENERAL ADVANCED MODULE (APPENDIX 3)

GA1 Foundations of Programming and Software Development
GA2 Advanced Elements of Programming GA3 Applications of Modelling

VOCATIONAL ADVANCED MODULE (APPENDIX 4)

VA1 Business Information Systems
VA2 Process Control Systems
VA3 Project Management

Core Module

UNIT C 1—HARDWARE

This unit should be done on a need-to-know basis during the teaching of other units.

Objective

Students should be able to identify and understand the functions of the main components of a typical computer system as well as identify and understand the functions of various peripherals.

Sub-objectives

Students should be able to:
1. identify the main components of the hardware in use (i.e. CPU, input devices, output devices and storage devices).
2. demonstrate an understanding of the functions of the main components of the hardware in use;
3. identify various peripheral devices (e.g.modem, fax-modem, plotter, scanner);
4. demonstrate an understanding of the functions of the various peripheral devices;
5. demonstrate an understanding of the network in use (if applicable).

Context

Students should fully understand how computers work and that the computer is under their control. They should not be mystified by computers and should be able to understand that computers are continually being improved and why. Students should also realise that any knowledgeable and interested person can build their own computer from components.

Content

Students should be able to differentiate between the basic components of a computer system and understand the function of various peripheral devices. If peripheral devices are not available for demonstration on site, students should visit facilities which have them.

Resources

Minimum necessary resources: Diagrams, models of the basic computer components; illustrations of their functions; actual samples or illustrations of peripheral devices.

Optional extra resources: Videos, visits to computer facilities, computer building kits; Software such as Computerworks; Dated, but not obsolete computer to take apart.

Links

All other units.

Methodology

Explanations with diagrams, video and the real objects with field trips where necessary Hands-on experience of building a computer model or a peripheral device.

Many elements of the IT curriculum can be embedded in existing subject areas (page 7) e.g. languages (see also page 78)

UNIT C2—SYSTEMS SOFTWARE ENVIRONMENT

This unit should be done on a need-to-know basis during the teaching of other units.

Objective

Students should be able to: understand the main functions of the systems software environment and to utilise Its features in relation to the main applications software being used.

Sub-objectives

Students should be able to:
1. demonstrate an understanding of the main functions of the system software environment
2. use the features of the system software environment (to the appropriate level) in relation to the main applications software
3. use network functions (if available) to the appropriate level.

Context

This unit is intended to familiarise students with the basic operating system of a computer (on a need-to-know basis) so that they can use the system competently to achieve their tasks.

Content

Students should know what system software is and the types available for various makes of computers. Students should also know the various steps and commands needed to perform a variety of tasks such as formatting a disk, copying a disk, making directories and sub-directories, hard-disk management, unformatting, saving and renaming files. Where applicable, students should also be familiar with the appropriate functions of the local or wide area network available to them.

Resources

Minimum necessary resources:
Computer, system software.

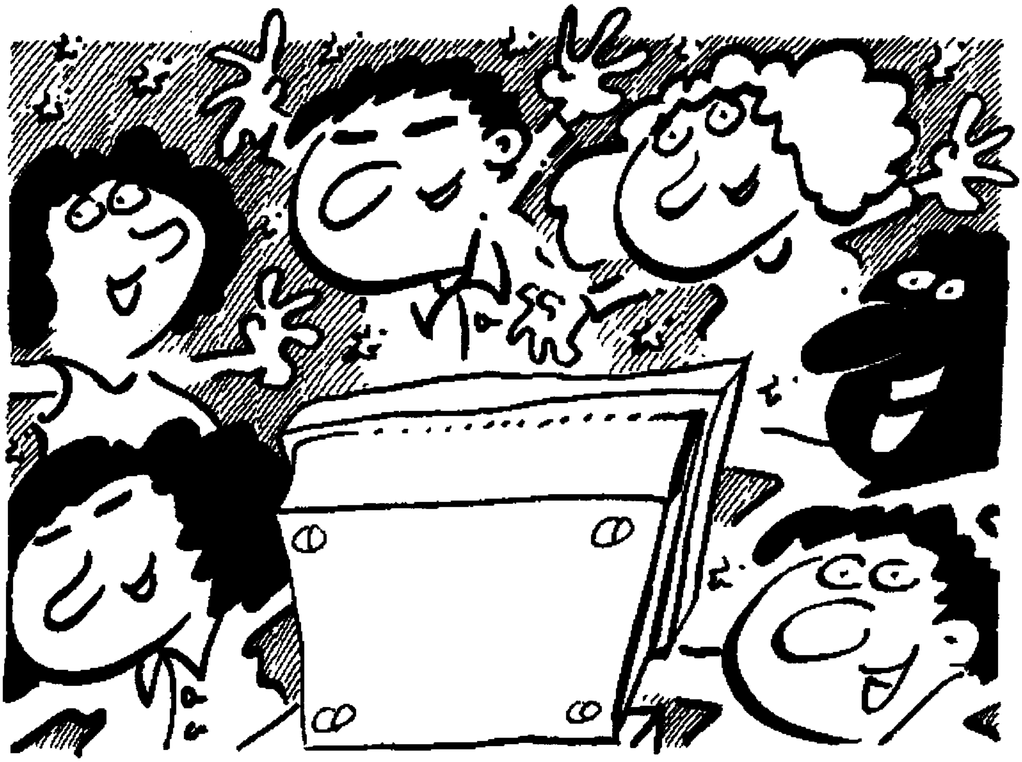
Optional extra resources:
Easy to understand manuals on the use of system software;
Tutorial software such as Teach yourself DOS, DOS Tutor.

Links

All other units.

Methodology

Demonstration of each system software command.



Information technology (IT) is one of the basic building blocks of modern society (page 5)

UNIT C3—COMPUTING TRENDS

This unit should be done on a need-to-know basis during the teaching of other units.

Objective

Students should be able to explain the current situation and trends in computing against the background of past developments.

Sub-objectives :

Students should be able to explain the current situation and trends in computing against. the background of past developments in: a) hardware; b) software; c) operating methods.

Context

Where are we going ? In order to answer the question we have to know where we have come from. The world of computing will be very different by the time students enter the work place, but many of the changes are predictable if we study the trends up to now. In addition, we have to know some of the history of computing in order to understand the terminology and procedures we find today.

Content

Students: should understand the key stages in the evolution of computers over the years. This may be looked at from the following points of view: early history (weaving, calculating machines, code breaking), CPU development (improvements in speed and power versus decrease in price, size and energy consumption); input devices (developments from punched cards to mice and speech recognition); output devices (from teletype to video display unit); and storage devices (from punched paper to hard disks), software (from changing the wiring to user-friendly software tools); text and document processing (leading to the ′paperless office"), and operating methods (developments from batch processing and time-sharing to local and wide area networks, multi-tasking and distributed processing).

Resources

Minimum necessary resources: Pictures or illustrations of earlier computers.

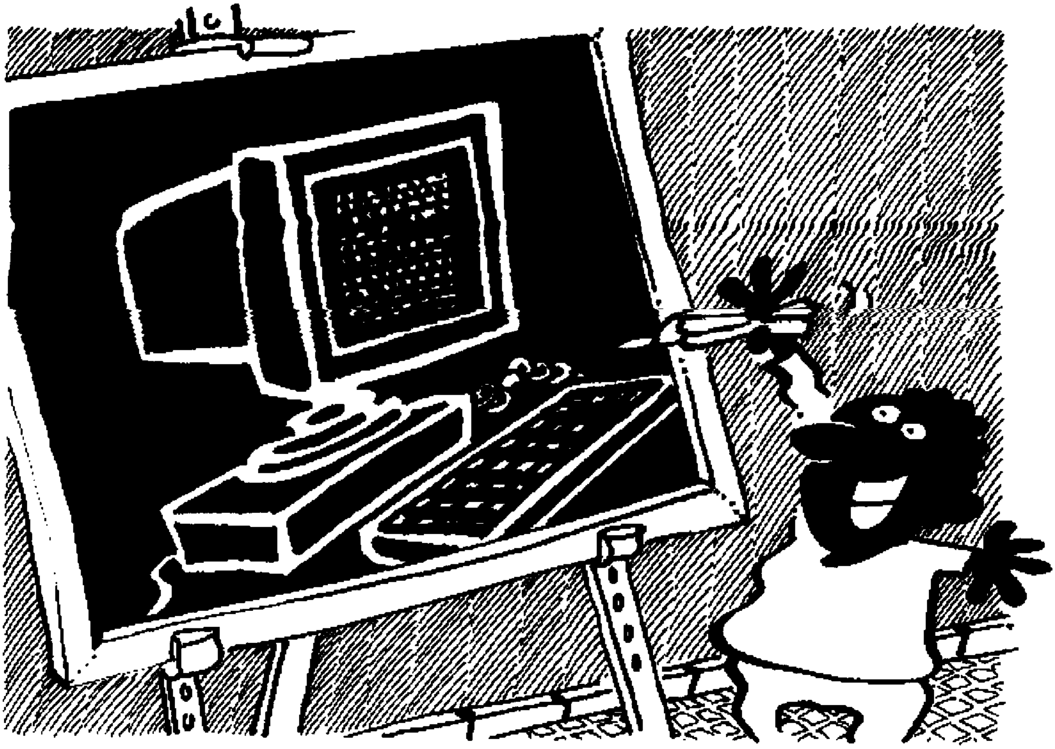
Optional extra resources:
Suitable books, newspaper clippings of newly launched computer hardware or software, newspaper articles of soon-to-be launched or future hardware and software, videos, examples of hardware, software and peripherals.

Links

All other units.

Methodology

Student-based research;
Visits to facilities with earlier and recent computer hardware.



Information technology (IT) is one of the basic building blocks of modern society (page 5)

UNIT C4—INTRODUCTION TO USING A COMPUTER

This Unit is for students who have never used computers.

Objective

Students should be able to show competence in using a computer to generate simple things such as posters, banners, signs, invitations cards, calendars and drawings.

Sub-objectives

Students should be able to:
1. demonstrate the ability to use a computer competently to produce posters, banners, signs, invitation cards, calendars and letterheads using simple software suitable for beginners;
2. experience the enjoyment and stimulatlon in using computers.

Context :

This unit is aimed at teaching students how to use a computer system. The student's first experience in using a computer should be fun and stimulating. Educational games, good CAL packages and simple graphics software can give students a good introduction to using a computer for the first time.

Content

Teachers should give a meaningful and directed exercise so that students have a definite objective to aim for. If necessary, teachers may illustrate the functions of the various components of the computer in terms of input processing, output and memory. Students should know howt to operate a computer system and its peripherals as well as the commands necessary to use the software to produce the required outcomes.

Resources

Minimum necessary resources: Computers for small group work; easy to use software such as Printmaster, PrintShop Bannermania or KidPix Printer.

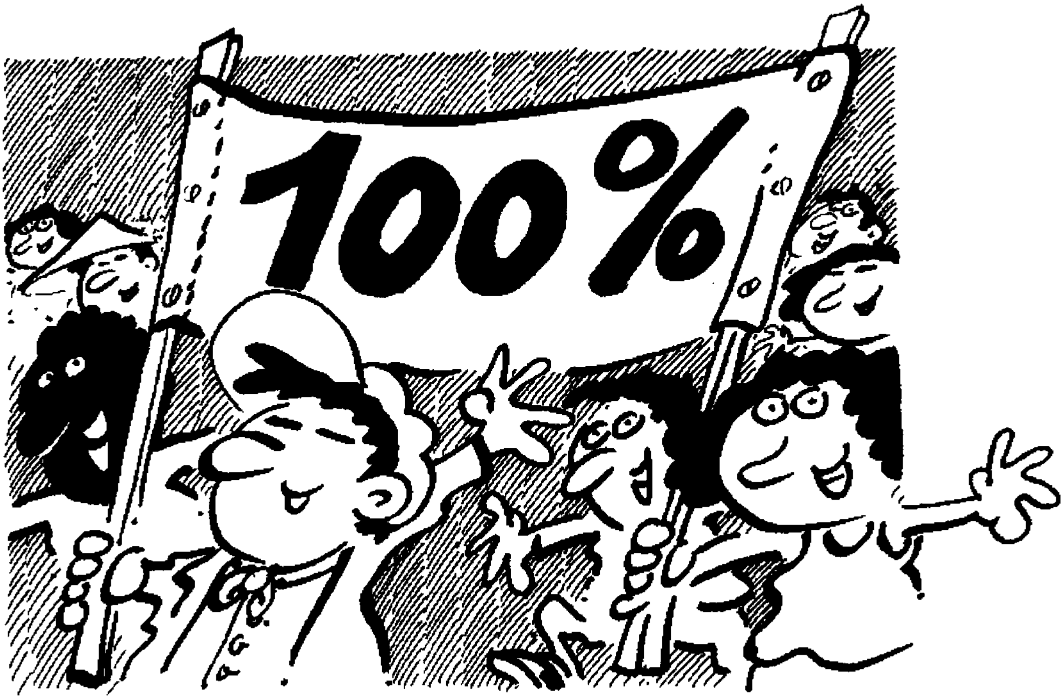
Optional extra resources: Samples of various creations using the software chosen; Other software such as Storybook Weaver, My Story.

Links

All other Units.

Methodology

Student-centred activities.



All secondary school children in all countries need a grounding in information technology (page 5)

UNIT C5—TEXT PROCESSING

Objective

Students should be able to use a word-processor skillfully and intelligently to produce various readable and structured documents for several disciplines.

Sub-objectives

Students should be able to:
1. produce a readable document
2. produce a structured document;
3. produce various documents;
4. discuss the advantages and disadvantages of a word-processor in a variety of settings; 5. make informed decisions as to whether a word-processor should be used for a given task.

Context

Being able to use a word-processor is necessary in today's society. Few people use typewriters when a computer provides a better alternative. There are clear advantages of using a word-processor compared to the usual method of writing on paper or typing with a typewriter. Students should appreciate the use of a word-processor and be encouraged to use it for most writing tasks. The objective is not to train a qualified typist or secretary, but a knowledge of word-processing and keyboard skills is an advantage when seeking employment.

Content

Students should first learn how to use a word-processor under the supervision of a teacher who should demonstrate and emphasize how easy it is to use. Students should start by typing in simple but meaningful exercises. They should know how to use the various features (e.g. bold, italics, underline, justified margins, centring, superscript, subscript, fonts, headers and footers) provided by a word-processor and, where possible, be able to use additional utilities such as spell checkers, grammar checkers, dictionary, thesaurus and merge facilities.

Meaningful activities on a word-processor include the preparation of personal or business letters invitations to school events, and lists of school events. Students should be able to use a word-processor independently to produce various documents which are readable and structured in a most presentable form. They should also be able to make informed decisions about whether or not using the word-processor for a certain task is the most efficient method.

Resources

Minimum necessary resources:
One computer per student, word-processing software;
Teacher prepared materials (exercise sheets, sample files).

Optional extra resources: Easy to understand manuals on word-processing software;
Liquid crystal display panel on overhead projector, magazine articles on word-processors;
Advertisements and brochures on word-processors available in the market.

Links

Hardware, Systems Software Environment, Choice of Software Tools, Computing Trends, Introduction to Using a Computer, Keyboarding Skills, Desk Top Publishing, Computers and Communication.

Methodology

Teachers may initially create simple exercises such as sample documents on disks, and require students first to open, modify and re-save files; then to progress to more difficult exercises such as use of headers, ' footers, dictionary, thesaurus, spelling and grammar checkers.

Unit C6—Working with a Database

Objective

Students should be able to use a prepared database in a competent manner.

Sub-objectives

Students should be able to:
1. understand some of the relevant phases of problem solving;
2. identify a problem which can be solved by a database
3. use a prepared database to store information
4. drawn and interpret information from an existing database in a structured and rational manner.

Context

In a large variety of business and in every day life, data are stored in databases managed by computers which are increasingly being linked together, for example for airline reservations and hotel reservations. More and more information about people is stored in databases—students should be aware of the need for the protection of personal data.

Content

Students analyze different applications in every day life where databases are used. Whenever possible, examples should be used which make the need for data protection obvious, such as: student records, information about tourist flows and needs.

Using a suitable example, the teacher prepares the appropriate structure of a database. Students then collect the necessary data, for example by interviews for which they must design an appropriate questionnaire. The data are entered into the database. Different lists will be produced and discussed.

During the process of using a database, students will come to understand some aspects of problem solving such as design, data entry and modification, and how to question the database.

Resources

Minimum necessary resources:
One computer per group of students;
Simple databases contained within "Works" packages
Some countries provide simple educational databases or special user interfaces for students at a reasonable price.

Optional extra resources:
One computer per student tutorial software
Liquid crystal display panel and overhead projector; Cardex file.

Links

All other Core Units; Core Elective Unit Database Design and Use is a next step.

Many elements of the IT curriculum can be embedded m existing subject areas (page 7) e.g. science (see also page 77)

UNIT C7—WORKING WITH A SPREADSHEET

Objective

Students should be able to understand and make use of a prepared spreadsheet.

Sub-objectives

Students should be able to:

1. demonstrate an understanding of what a spreadsheet is and its uses;
2. use a prepared spreadsheet to change values of variables and to see its various effects;
3. demonstrate an understanding of the uses of a spreadsheet.

Context

Spreadsheets are useful tools for individual or group work, and are widely used in industry and commerce. Students should understand what a spreadsheet is, how easy it is to manipulate variables, and should see the effects of this manipulation.

Content

Students are introduced to the concept of a spreadsheet and the components in a spreadsheet. They manipulate a spreadsheet by changing values in cells, and will also perhaps change formulas to see the effects. Graphs, based on the values entered in the cells, may be generated automatically,. Students should also be able to understand the various uses of a spreadsheet in daily tasks.

Resources

Minimum necessary resources: Computer systems for group work, spreadsheet software; Teacher-prepared spreadsheets.

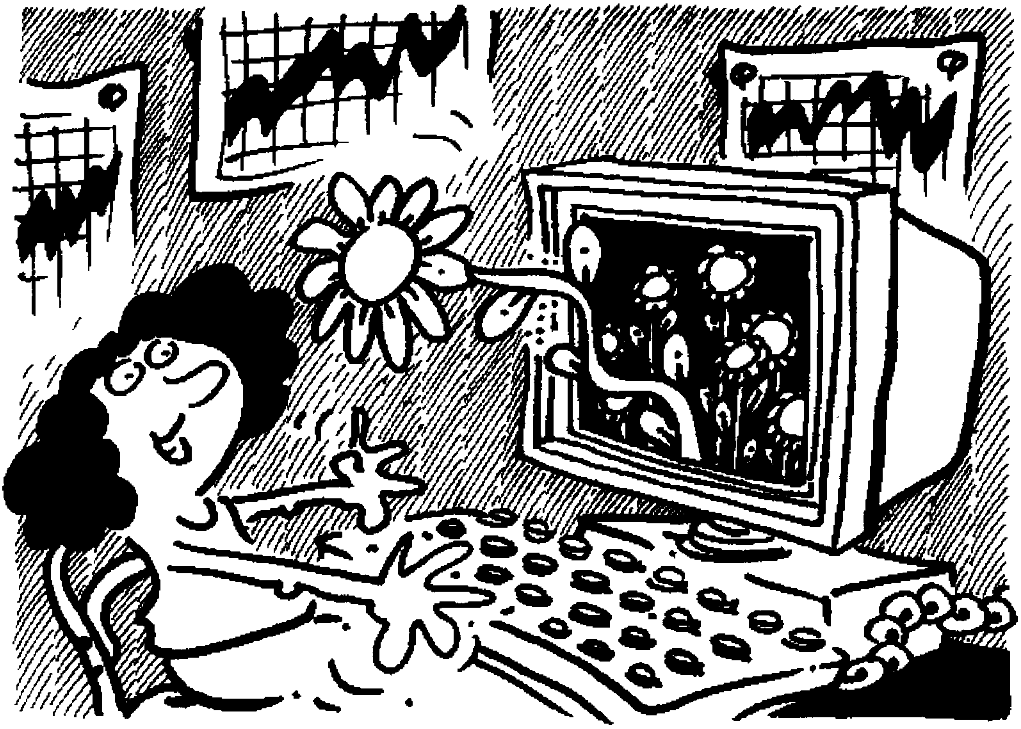
Optional extra resources: Ready-made spreadsheets; Liquid crystal display panel and overhead projector; tutorial software.

Links

All other Core Units; Core Elective Unit Spreadsheet Design and Use is a next step.

Methodology

Demonstrations, hands-on activities.



Many elements of the IT curriculum can be embedded in existing subject areas (page 7) e.g. science (see also page 77)

UNIT C8—WORKING WITH GRAPHICS

Objective

Students should be able to use graphical representations in an appropriate way.

Sub-objectives

Students should be able to
1. identify the use of different forms of graphic representations in every day life;
2. understand the link between data and graphics;
3. convert data into appropriate graphic representations;
4 summarise data into appropriate graphic representations;
5. use appropriate graphic representations to illustrate a document;
6. see the consequences of different forms of graphic representations of the same data;
7 identify cases where it is appropriate to use graphics, and select which graphics to use.

Context

"A picture can tell more than thousand words". A large variety of reports are illustrated by different graphic representations, for example, line graphs, bar charts or pie charts. Data can be shown in appropriate or inappropriate ways; the latter leading to misinterpretation of the original data. Familiarity with the ways in which graphs are used will help students to present their work clearly in many subject areas and in their working lives.

Content

Students analyse different application in every day life where graphic representations are used. Whenever possible examples should be used which make an inappropriate use obvious. Examples might be found in the school environment, in data about the community, in the business world, and in the daily or weekly newspaper. Current data and data from earlier years should be illustrated in different forms.

Resources

Minimum necessary resources: One computer for each group of students with a graphics software package; Graphics modules are included in all "Works" packages and with most spreadsheets; Examples from industry and commerce.

Optional extra resources: Advanced graphics software; Liquid crystal display panel for overhead projector.

Links

All other Core Units; Optional Unit Creating Graphics can be a next step.

Many elements of the IT curriculum can be embedded in existing subject areas (page 7) e.g. speech recognition in languages (see also page 79)

UNIT C9—SOCIALAND ETHICAL ISSUES

Objective

Students should be able to understand the social, economic and ethical issues associated with the use of computers.

Sub-objectives

Students should be able to demonstrate an understanding of:

1. the benefits and drawbacks of computer use to society in general;
2. the economic advantages and disadvantages of the use of computers;
3. the ethical questions which have arisen as a result of computer use.

Context

Students should be made to realise that computers do not always contribute positively to society. They should appreciate the seriousness of the social, economic and ethical issues over the years. There can be misuses and abuses as well as unethical behaviour by those in control of computing facilities. Students should be aware of such behaviour and how it can be corrected.

Content

Students are expected to understand basic concepts such as computer crime and fraud, equity intellectual ownership, privacy of information, links between automation and unemployment, and computer security (theft, hacking, viruses).

Resources

Minimum necessary resources: None.

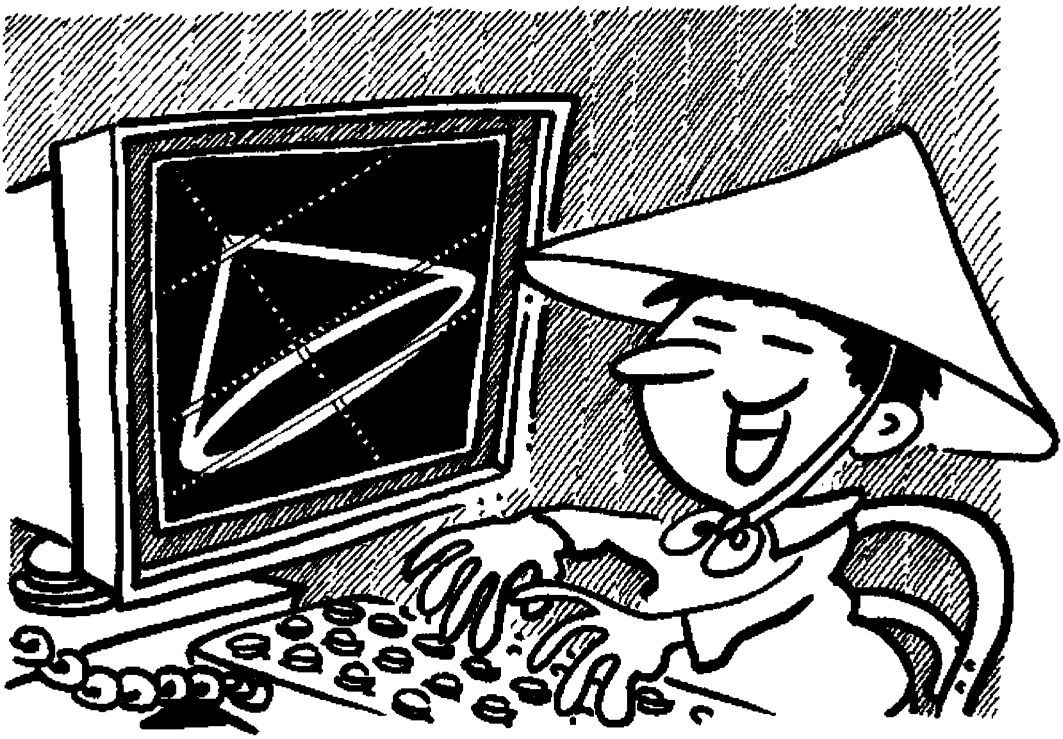
Optional extra resources: Newspaper clippings of stories of computer crime.

Links

All other Units.

Methodology

Discussions. or research by individuals or groups of students.



Many elements of the IT curriculum can be embedded in existing subject areas (page 7) e.g. mathematics (see also page 78)

UNIT C10—CHOICE OF SOFTWARE TOOLS

Objective

Students should be able to analyze different problems in order to decide which software tool is the appropriate one to use.

Context

During a Computer Literacy course, students get to know different types of software tools. By the end of the course they should be able to select the best tool available for a given task.

Content

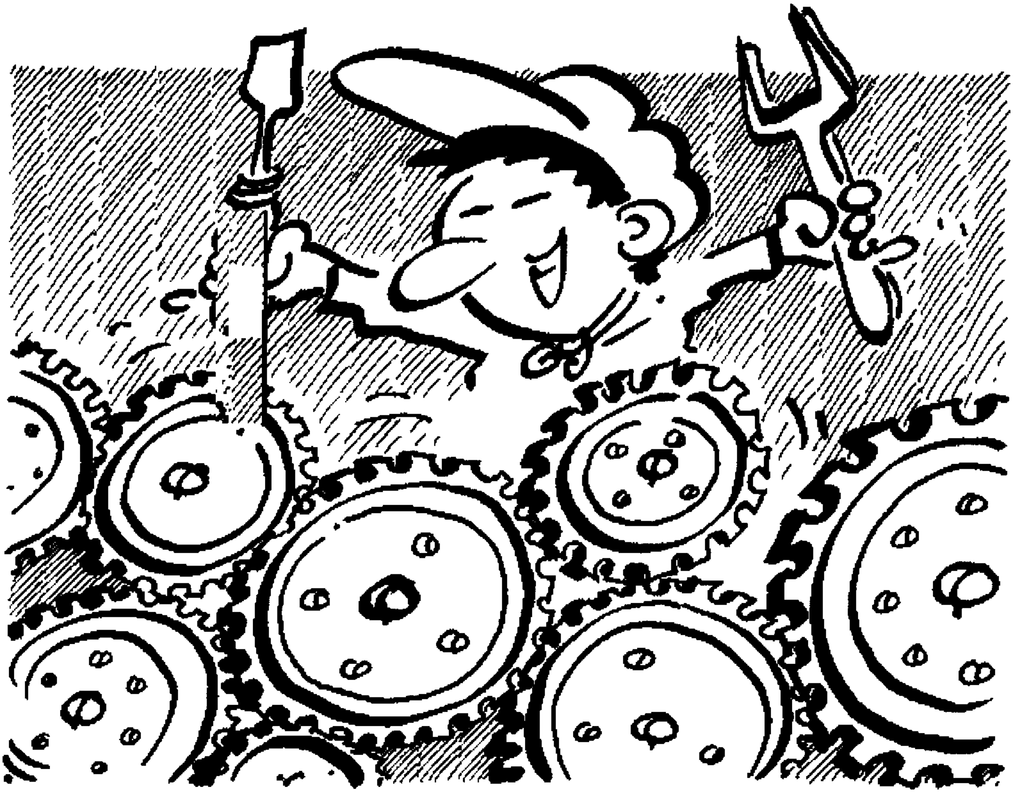
This unit might be done in two ways: (a) the students must identify the most suitable tool available for a given problem; or (b) the students must provide examples of other applications for tools they know. It should be obvious that there is an overlap between available tools, for example information may be stored either in a database or a spreadsheet.

Resources

No special resources are needed.

Links

All other Units.



Different countries will be at different phases of development:
Automation Phase (pages 7, 17, 19)

Core Elective Module

UNIT E1—DATABASE DESIGN AND USE

Objective

Students should be able to create and use databases in a competent manner.

Sub-objectives

Students should be able to:
l. understand some of the relevant phases of problem solving;
2. identify a problem which can be solved by a database;
3. design and create an applicable database; 4. obtain data and enter it into a database
5 draw and interpret information from a database in a structured and rational manner; 6 understand the principles of personal data protection.

Context

This Unit extends the experience gained in studying Optional Unit Working with a Database. Familiarity with the main commercial databases is an advantage when seeking employment.

Content

Using a suitable example, students design an appropriate questionnaire and collect data. They students make a first structure of a database and enter the data. Some fields might be wrong and may have to be modified. The database can be restructured until a usable database has been established.

During the process of using the database, students will become familiar with some of the main aspects of problem solving: design, data entry and modification, as well as use of the application Principal data types such as text and number will be covered. If systems with a programming language like dBASE are available, principle elements of programming, such as conditions and loops, can also be covered.

Analyzing the use of databases will demonstrate how commercial processes operate for example there would be a serious problem for a new airline if it were not allowed to use the existing reservation system. Students should also be familiar with very large databases (VLDB) and Geographic Information Systems.

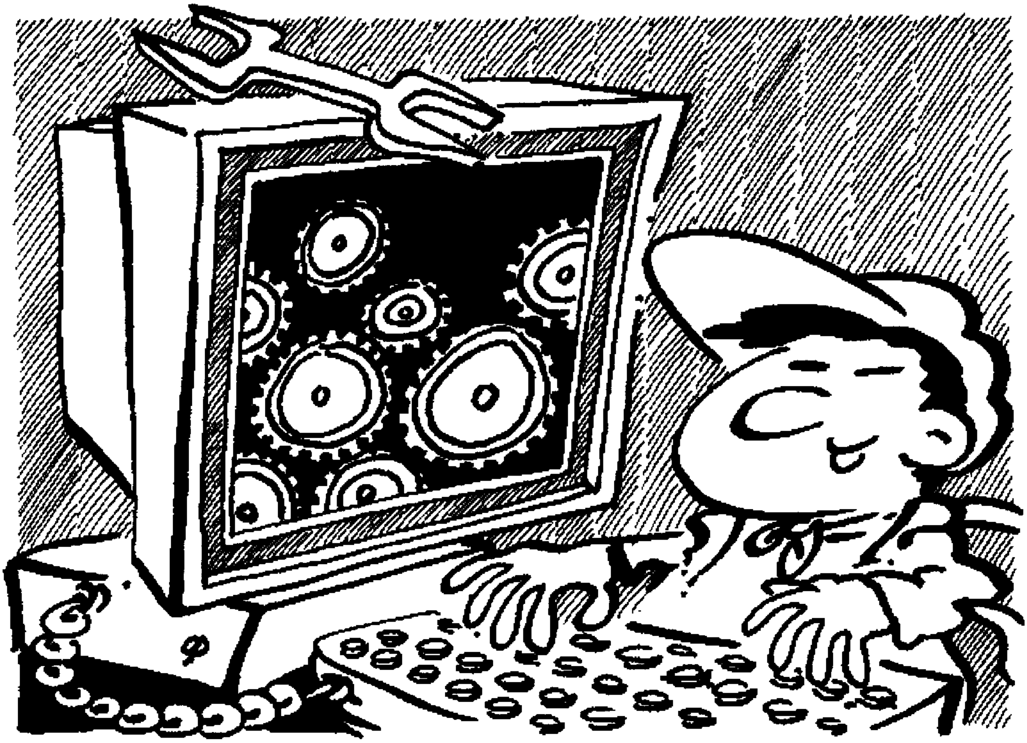
Resources

Minimum necessary resources:
One computer per group of students;
Simple databases contained within "Works" packages;
Some countries provide simple educational databases or special user interfaces for students at a reasonable price.

Optional extra resources: Advanced database such as dBase; Liquid crystal display panel for overhead projector.

Links

This is an extension of the Core Unit Working with a Database.



Different countries will be at different phases of development
Information Phase (pages 7, 17, 19)

UNIT E2—SPREADSHEET DESIGN AND USE

Objective

Students should be able to design and create a spreadsheet to solve a given problem.

Sub-objectives

Students should be able to:
1. identify the problem that can be solved by a spreadsheet;
2. enter and lock headings of columns and rows;
3. vary width of columns
4 allocate values to columns
5 create correct formulas;
6. copy formulas;
7. re-arrange entries (rows);
8. re-arrange columns;
9. import information from a variety of sources;
10.export information to a variety of destinations.

Context

The ability to create and use spreadsheets has become essential for all involved with financial managements research, and many other commercial activities. Spreadsheets have the advantage that most can easily accept and manipulate information from databases and can create output as graphics. The ability to use spreadsheets efficiently is a decided advantage when seeking employment.

Content

Creating a class hobbies list may be a good first exercise. Students should then create at least one spreadsheet to draw up a budget in which a number of "what-if' situations arise. If the spreadsheet allows it, the reverse of this process should also be utilised.

Students should create a simple spreadsheet to accept and calculate the results of an experiment or an opinion survey in at least one other discipline. A number of spreadsheet applications will arise naturally in most school activities. Spreadsheets should be used to create lists which can be re-arranged in a different order as needed.

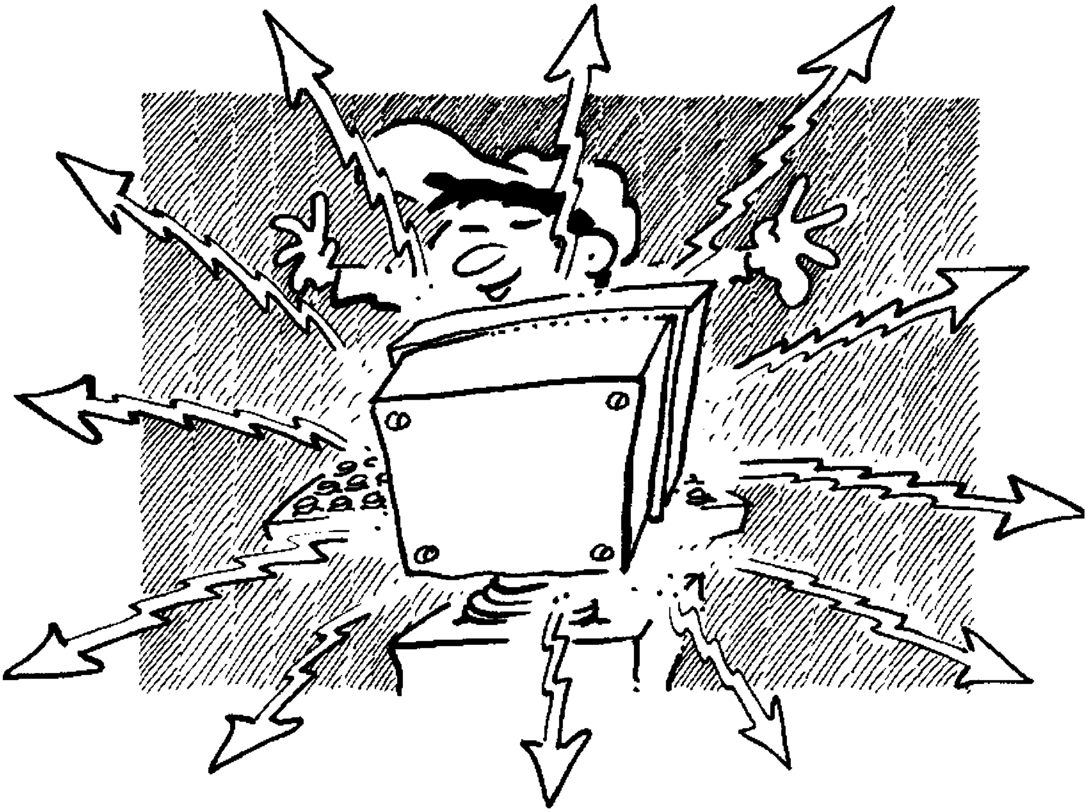
Resources

Minimum necessary resources:
Simple spreadsheet program;
One computer for a group of students.

Optional extra resources:
Printer;
Training video or computer program;
Liquid Crystal Display panel and overhead projector.

Links

Core Unit Working with a Database, Core Elective Unit Database Design and Use. This is an extension of the Core Unit Working with a Spreadsheet.



Different countries will be at different phases of development:
Communication Phase (pages 7, 18, 19)

UNIT E3—CAREERS IN INFORMATICS

Objective

Students should be able to describe career opportunities in the computer field, together with the required qualifications.

Context

The titles and job descriptions of Information Technology (IT) personal have grown out of IT history and are not self-explanatory, even to the computer literate person. Many students will contemplate a career in IT. They should be helped to make an informed choice.

Content

Students should research and report on the careers available in the computer industry, including systems development and the provision of services.

Resources

Minimum necessary resources: Careers opportunities library

Optional extra resources: Presentation packages; Projection facilities.

Links

Core Module Computing Trends.

Methodology

This may be a good opportunity to send students or groups of students out to interview IT practitioners and to interpret the information gathered. The use of a word-processor and a presentation tool should be appropriate to illustrate the results of research. Qualification survey graphs could also be created. A spreadsheet could be used to list and compare variables, for example investment in training versus potential income.

Many units are of special interest to students about to enter the workplace e.g. keyboarding skills (pages 12, 51)

General Option Module

UNIT OP1—KEYBOARDING SKILLS

Objective

Students should be able to use a keyboard efficiently and effectively.

Sub-objectives

The student will be able to:
1. find all keys without delay;
2. use the correct fingers for eventual touch-typing.

Context

An inability to use a keyboard efficiently will eventually make computer use a frustrating experience for students. Incorrect typing procedures will place a limit on the speed students can achieve, and will make the learning of correct typing procedure more difficult. The teaching of blind typing is not an objective of this course. Teachers may want to teach keyboarding skills as a unit on its own, or as part of other units.

Content

Games may be used to familiarise students with the keyboard; including tutorial programmes to teach typing skills and computer assisted learning activities requiring minimal keyboard input. Simple practical programming in beginners' languages such as Logo or Boxer can help to prepare for later use of computer languages. Word-processor exercises and simple desk top publishing will give meaning to these first technical exercises and lead on to a fuller treatment as described in other Units.

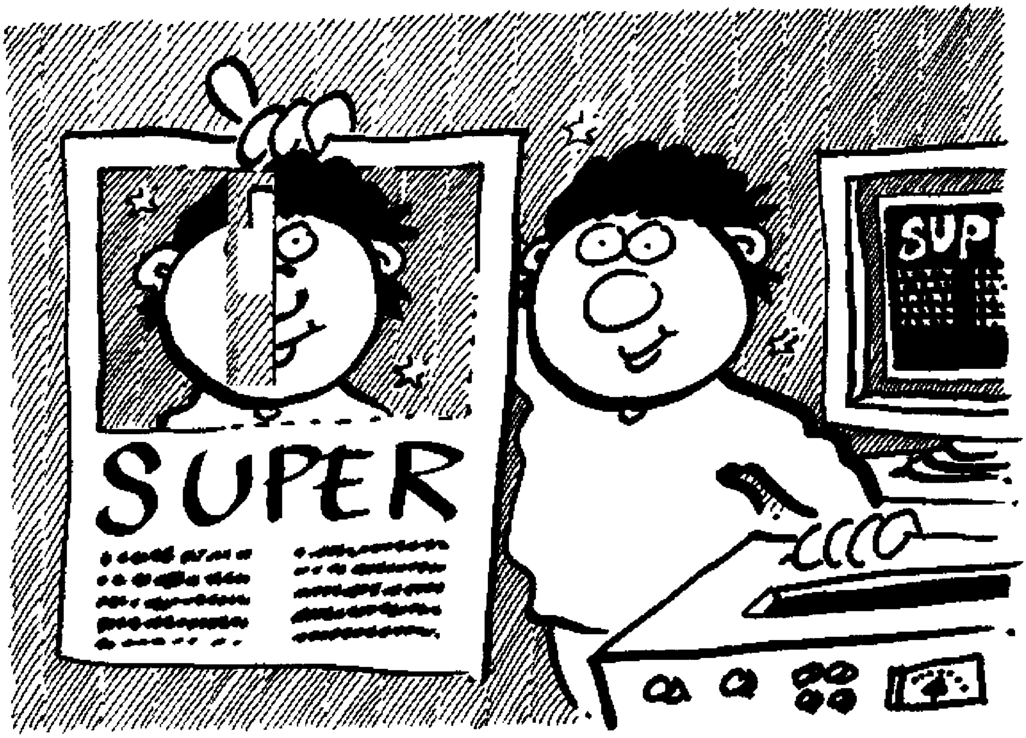
Resources

Minimum necessary resources: One computer per student.

Optional extra resources:
Commonly available typing handbook; suitable typing skills programme, such as Mavis Beacon Teaches Typing, or Typing Tutor;
Video on typing skills.

Links

Core Units; Desk Top Publishing.



Many units are of special interest to students about to enter the workplace e.g. desk top publishing (pages 12, 53)

UNIT OP2—DESK TOP PUBLISHING

Objective

Students should be able to understand the use of desk top publishing software and use it proficiently to produce professional looking documents.

Sub-objectives

Students should be able to:
1. demonstrate an understanding of a desk top publishing software;
2. convert a text file from a word-processor into a functional file for the desk top publishing environment;
3. produce a professional looking document with the help of a desk top publishing software.

Context

Although the use of desk top publishing (DTP) software is not as essential as the use of a word-processor, it is nevertheless, an advantageous tool to use. Students should be able to appreciate the advantages of using DTP software to produce professionally-looking documents. This should motivate students to want to use DTP where suitable to create such documents.

Content

Students should understand the basic terminologies used in DTP such as columns, kerning, fonts, font sizes and mastheads. They should also be able to use DTP software to convert an existing word-processor file into a format suitable for use in a DTP environment.

Students should be taught how to manipulate this file in such a way as to produce an attractive document. Students could be asked to create school newsletters with at least two columns; and brochures for school events.

Resources

Minimum necessary resources: One computer per student; DTP software Teacher-made samples of documents produced with DTP software.

Optional extra resources: Samples of professionally-produced brochures, pamphlets, documents created with DTP software; Easy to understand DTP software manual.

Links

All other Units.

Nothing is more motivating for students than communicating with a native speaker of a foreign language in another country (page 55)

UNIT OP3—COMPUTERS AND COMMUNICATION

Objective

Students should be able to demonstrate an understanding of how the combination of computers with telecommunications has facilitated a faster and more versatile way of communication.

Sub-objectives

Students should be able to: 1. identify the various methods of communicating via computer 2. demonstrate an understanding of electronic funds transfer 3. send and receive messages by using an electronic mailing (e-mail) facility; 4. communicate with bulletin board systems by being able to read and leave messages, upload and download files 5. send, receive, read and print faxes using a fax-modem and an appropriate software too].

Context

In an information society, it is increasingly necessary to get information quickly from an appropriate source and to exchange information speedily with others throughout the world. Students should know the various means available and how to access those means.

Content

Students should understand the various means of electronic communication such as electronic mail, use of electronic bulletin boards, faxing with the computer and modem; and should be able to make an informed choice as to which one is the best to use in a given situation.

Resources

Minimum necessary resources: One computer per group; communications software; Access to bulletin board systems E-mail and fax facilities.

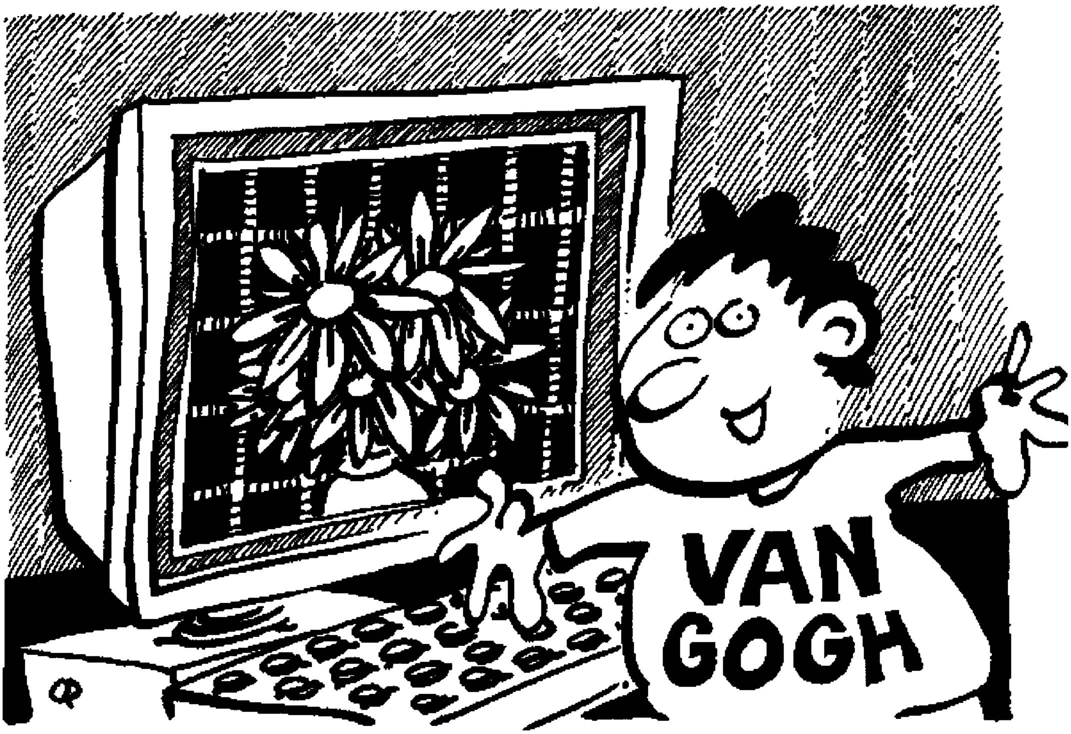
Optional extra resources: Liquid crystal display panel and overhead projector; Diagrams and illustrations.

Links

All other Units.

Methodology

Hands-on experience.



Graphics packages allow for the creation of original artwork (pages 57, 80)

UNIT OP4—CREATING GRAPHICS

Objective

Students should be able to use suitable graphics tools to create the graphics required for a variety of purposes.

Sub-objectives

Students should be able to:
1. analyze a task and select the best graphics tool from the tools available
2. import drawings (graphics) from other sources, using a scanner if available;
3. use graphics software to draw and animate drawings from other sources;
4. produce attractive and suitable graphics-based materials.

Context

The world is visually based. It is important for students who wish to create visual materials to know how to use a computer and appropriate software to produce these materials.

Content

Students should know the various graphics software tools available and be able to choose the best one for a particular task. They should be able to use the software competently to create documents including graphics for a variety of other disciplines.

Resources

Minimum necessary resources:
One computer per student, graphics software

Optional extra resources: Various input and output devices, samples of various computer-produced graphics based materials.

Links

All other Units.

Methodology

Demonstrations, hands-on experience, projects.

Many units are of special interest to students about to enter the workplace e.g. working with multimedia (pages 12, 59)

UNIT OP5—WORKING WITH MULTIMEDIA

Objective

Students should be able to plan and execute an efficient multi-media presentation.

Sub-objectives

Students should be able to:

1. manage static components of multimedia like text with pictures;
2. manage dynamic components of multimedia like sound with animation;
3. manage the incorporation of sound and video from "outside" sources such as audio tape and video desk.

Context

Students should study the evolution of multimedia technologies (from analog to digital) and be able to identify the main components of a multimedia presentation (sound, text, pictures, animation). The ability to prepare a multi-media presentation is a valuable skill in today's job market.

Content

Activities are closely dependent on the availability of equipment systems and multimedia tools. These involve: sound (music), text (DTP); pictures (graphics); animation (graphics); and communication systems.

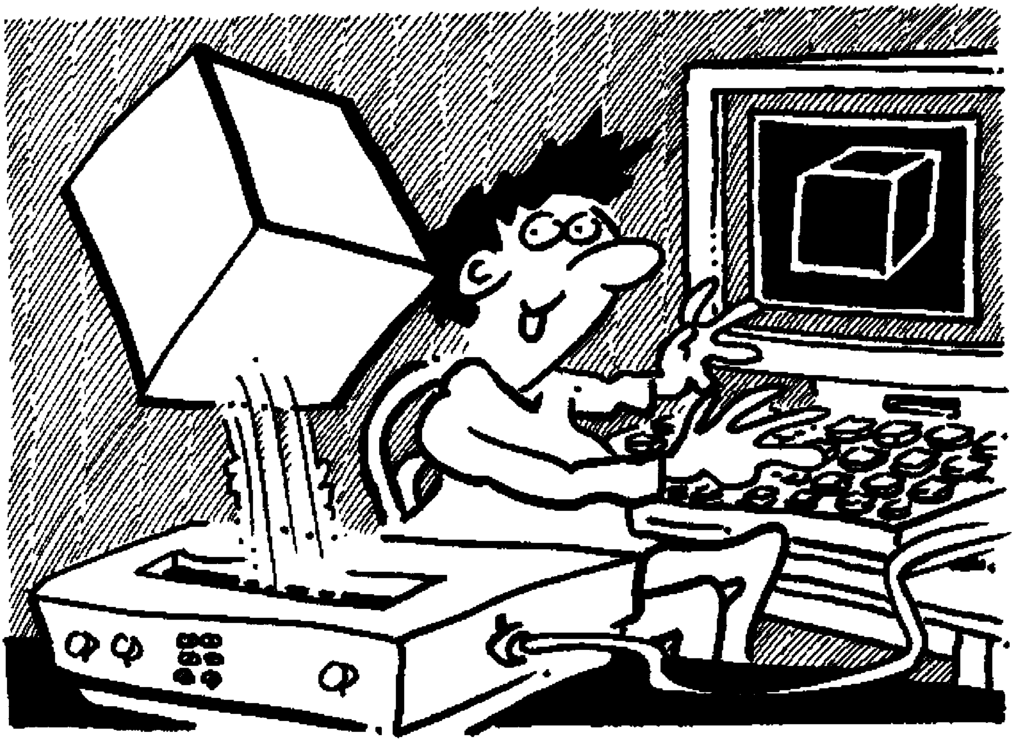
Resources

Minimum necessary resources:
Suitable computer with internal or external multimedia devices
Application software.

Optional extra resources:
Loudspeakers; video projector; latest professional software.

Links

All other Units.



Many units are of special interest to students about to enter the workplace e g. computer assisted design (pages 12, 61)

UNIT OP6—COMPUTER AIDED DESIGN (CAD)

Objective

Students should be able to produce such designs as are required by other disciplines, using a suitable Computer Aided Design (CAD) package.

Sub-objectives

These should be established in cooperation with the curriculum specialists for other disciplines. They are likely to include the ability of the student to perform operations including:
1. simple plan drawings and other two-dimentional drawings;
2. three-dimentional drawings;
3. creating special shapes and curves;
4. shading;
5. rotation;
6. inserting text.

Context

The CAD program is the word-processor of the drawing-board. It allows changes to be made without having to redraw the entire drawing, and it allows for elements of drawings to be used repeatedly. It is now impossible to imagine a modern design office without CAD. While CAD systems available in schools will not be as sophisticated as those in the work place, students should be introduced to the advantages of CAD where applicable. Experience even with only simple CAD may lead to employment
opportunities.

Content

Students involved with disciplines which require use of a drawing board should be given experience of doing the same work with a CAD program.

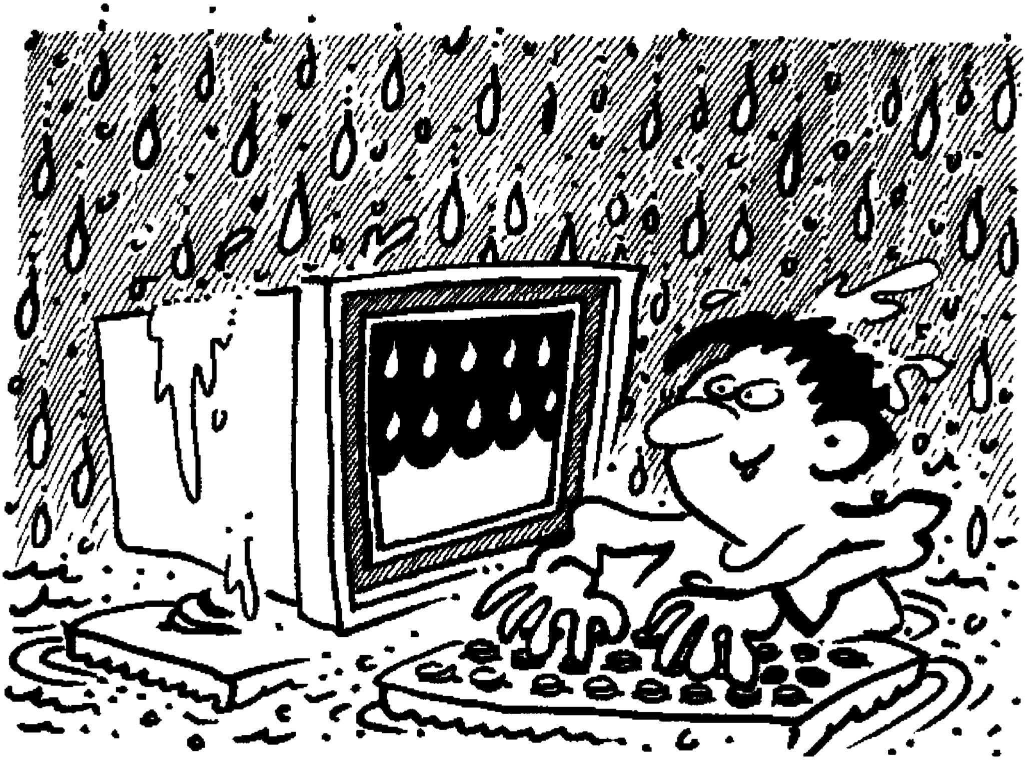
Resources

Minimum necessary resources:
One computer for each student;
A simple CAD package;
A suitable printer.

Optional extra resources:
Suitable plotter;
Large format screen;
Training video or program for the package.

Links

Spreadsheet Design and Use. The tables of quantity of materials which need to be created once the drawings are complete call for a spreadsheet. The link between size, quantity of material and budget also creates opportunities for advanced spreadsheet work.



Students should be able to simulate a real life situation, such as pollution of a river, landing an aircraft (page 63)

UNIT OP7—MODELLING AND SIMULATION

Objective

Students should be able to identify the main parameters of a real situation, formalise a model, then explore it, interpret the results and determine how the model fits with reality.

Sub-objectives

Students should be able to:

1. use existing models, varying relevant parameters and interpreting results;
2. modify an existing model, varying relevant parameters and interpreting results;
3. model situations with a small numbers of parameters, explore the model and interpret the results.

Context

Often, when teaching and explaining open phenomena in school, presentations are made in an excessively deductive way. To tackle, even at an elementary level, the modelling of a simple situation, allows one to balance deductive aspects with an exploratory approach (simulation versus modelling). This unit offers the opportunity to solve some true problems instead of "artificial" ones. Such experiences bridge the experimental and the theoretical (formal) approaches.

Content

A good introduction would be to simulate and experiment with an already solved problem (an existing model). Examples include radio-active decay, change in Ph values, population changes. Published software examples are Lemonade, Flight Simulator.

Modifying an existing model, after running a simulation to try to understand the more important relation between the main parameters, helps to clarify the necessary basis for the real modelling process. Examples include supply and demand, pollution effects, running a company.

From concrete observations, very often visual ones, students can build up an outline of a system allowing them to reproduce the observed behaviour in an adequate way.

Students should be model on a spreadsheet and on a special modelling tool, if available.

Resources

Minimum necessary resources:
One computer per group of students; a spreadsheet program.

Optional extra resources:
Existing commercial software, such as Simearth, Simlife, Simcity;
Specific tools with graphical interfaces inspired by dynamic systems (Stella, Modus, Extend)
Specific tools dealing with numeric and symbolic calculations (Mathematica, Derive, Mathlab, Mapple).

Links

This Unit can be a first step towards the Advanced Unit Applications of Modelling.

Methodology

Although many different approaches are possible, depending on the choice and availability of tools, it is essential, when teaching such a unit at the Foundation Level, to limit oneself to simple models, even for complex situations.

UNIT OP8—EXPERT SYSTEMS

Objective

Students should be able to create and demonstrate the use of an expert system.

Sub-objectives

Students should be able to:

1. use expert systems created by others;
2. use suitable software to create such expert systems as may be of use in other disciplines;
3. identify situations where the use of an expert system is appropriate.

Context

Creating an expert system IS an excellent way of teaching. While the final product may be of limited use, the process of creating an expert system focuses the mind of the learner on the relevant issues—possibly like no other activity can.

Content

Various disciplines will provide different opportunities for expert systems. These systems may vary from being able to do simple tasks such as changing and classifying a regular shape, to such complex tasks as classifying botanical samples—or even predicting the weather; expressing the probability of various possibilities.

Resources

Minimum necessary resources:
One computer for a group of students;
Simple Expert System.

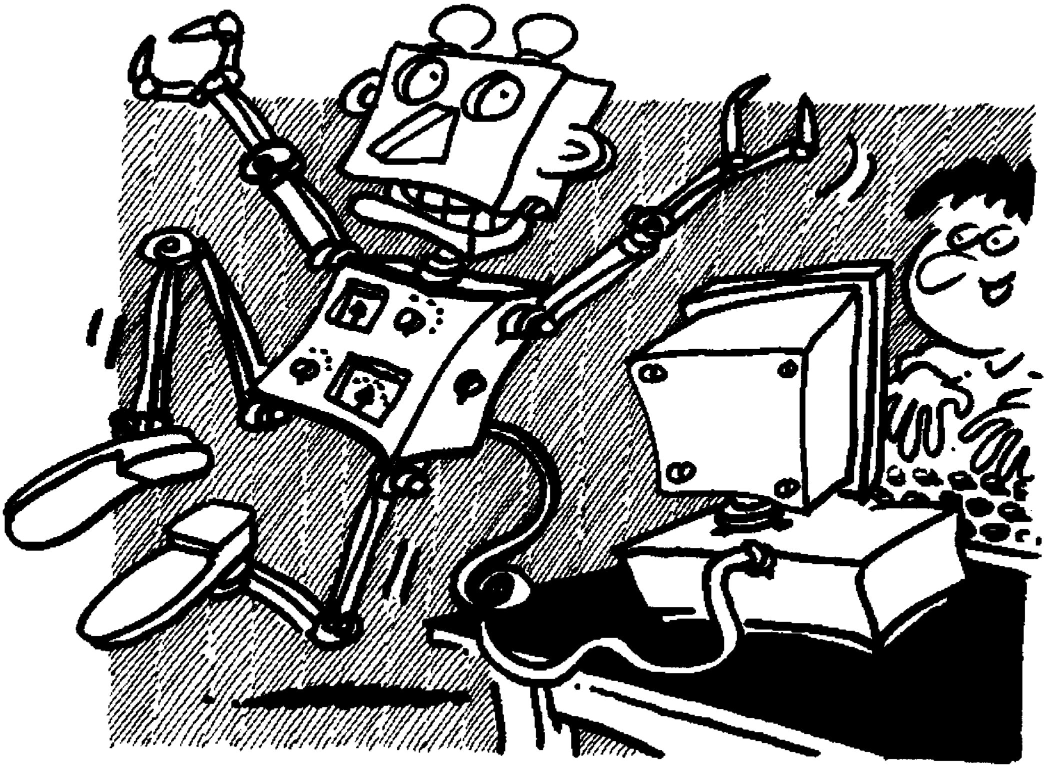
Optional extra resources:
Printer.

Links

Advanced expert systems usually allow the import of data from databases and spreadsheets. Where the output indicates probabilities, these can be presented in graphic form. Writing one's own expert system is an excellent programming task.

Methodology

Very suitable for working in pairs or groups. The arguments about the rules of expert systems are an excellent way of learning.



Students should be able to operate a simple robot (page 67)

UNIT OP9—ROBOTS AND FEEDBACK DEVICES

Objective

Students should be able to operate a simple robot and use a simple feed-back device.

Sub-objectives

Students should be able to:
1. control a specific robotic device
2. identify the situations where feed-back devices are useful;
3. use a simple feed-back device.

Context

This unit is a link to the industrial production in a country. The number of robots and feed-back devices is increasing rapidly with consequences for the labour market and the qualifications of people needed by employers. Students should have an understanding of the operation of robots, as they are frequently and increasingly being used in (a) dangerous situations, (b) improving the quality of the products, and (c) reducing labour costs.

Content

Students should use a simple software tool, or prepared user-interfaces, to control a robot. They should also attempt to build a simple robot device. v

Students should use feed-back devices in experiments in other subjects, such as a microphone in Physics or a thermometer or Ph-meter in Chemistry.

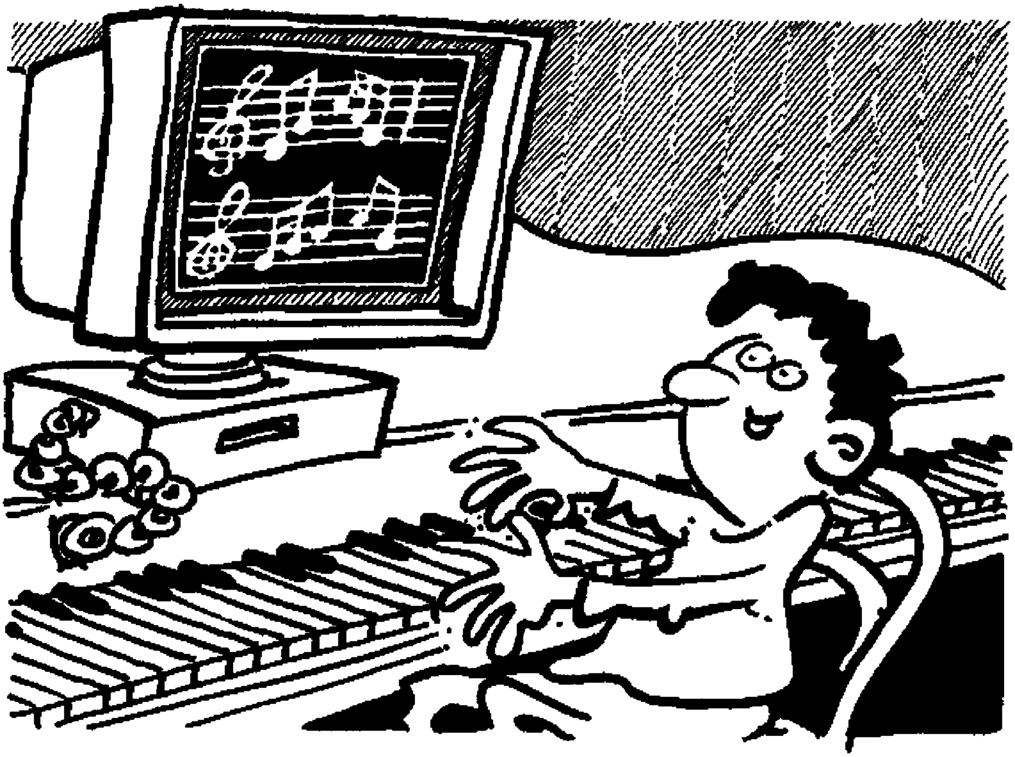
Resources

Minimum necessary resources:
A simple progammable robot device—with its own keypad or connected to a computer—for each group of students;
A simple feed-back device which provides input for a suitable computer programme.

Optional extra resources:
Robot, with control hardware and software;
Hardware and equipment to build robots at school Tools like LEGO-LOGO, Fischertechnik;
Several kits with the necessary hardware and software are available, such as the IBM and Brüderbund Labkits.

Links

Hardware, Computing Trends, Social and Ethical Issues, Programming.



Students should be able to create and play music using hardware and software (page 69)

UNIT OP10—MUSIC

Objective

Students should be able to create compositions, musical scores and arrangements of music as required by that discipline, using appropriate hardware and software.

Sub-objectives

To the level required by the discipline, students should be able to:
1. explore the use of appropriate hardware and software to play music, modify and print scores;
2. demonstrate the use of appropriate hardware and software to compose musical pieces;
3. demonstrate the use of appropriate hardware and software to arrange musical pieces;

Context

Available software enables musical scores to be written and edited with a computer in the same way that word-processing packages are used for composing text. Music can be played back and listened to via suitable hardware. Students who are required to write musical scores should realise that the computer facilitates the process. They should also know how to use appropriate software and take advantage of it for personal use.

Content

The teaching of music is not an objective of the informatics course. Before this unit, students should be able to identify the correlation between score and music. They should be able to modify an existing score in order to achieve a stated or desired effect. This should eventually be developed to the point where they compose original scores on a computer and arrange such compositions for different instruments, again using a computer.

Resources

Minimum necessary resources: Computer, music creation software.

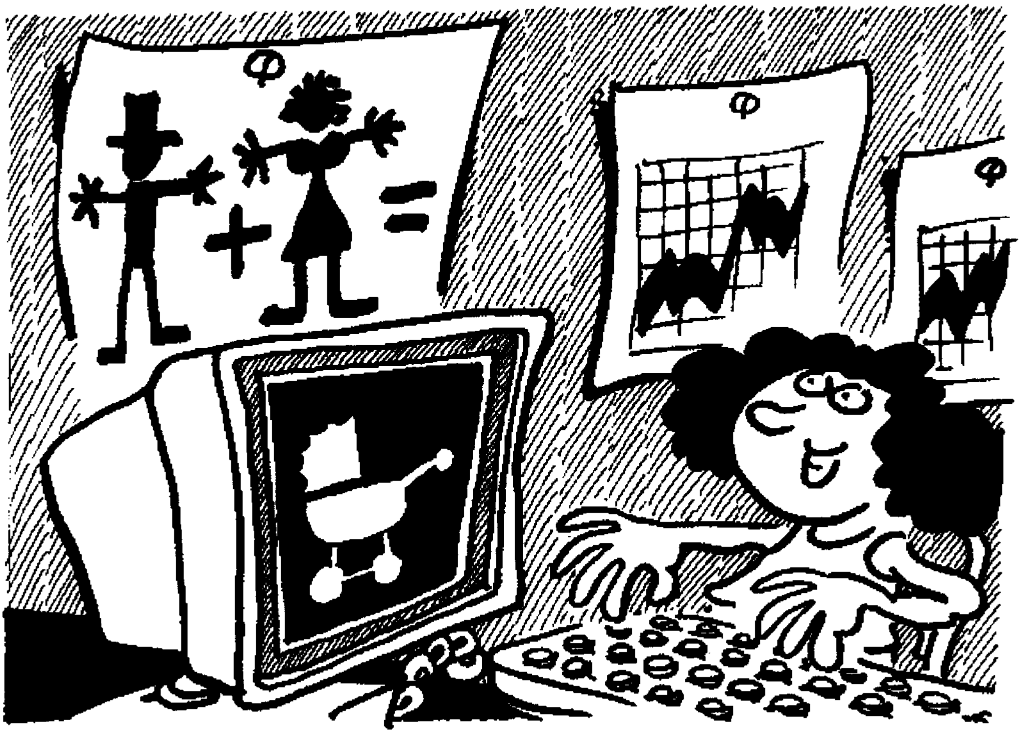
Optional extra resources: MIDI interface and associated hardware.

Links

Hardware, Systems and Software Environment, Keyboarding Skills, Text Processing.

Methodology

Hands on experience.



Students should be able to use a simple statistical package (page 71)

UNIT OP11—STATISTICS

Objective

Students should be able to use a simple statistical package to the level required by other disciplines.

Sub-objectives

Students should be able to: :
1. call up the correct section of a statistical package and enter data;
2. utilise output to the degree demanded by the specific discipline.

Context

The teaching of statistics is not an objective of the IT course. However, where students use statistics in other disciplines, they should be able to handle a simple statistics program and create an output which they can interpret in context.

Content

This unit is likely to be used in the senior secondary phase. Other disciplines most likely to use the Unit are economics, mathematics, agricultural science, biology and the social sciences. The complexity and volume of statistical work done will depend entirely on what is required by other disciplines.

Resources

Minimum necessary resources:
One computer per group of students;
Simple statistical program

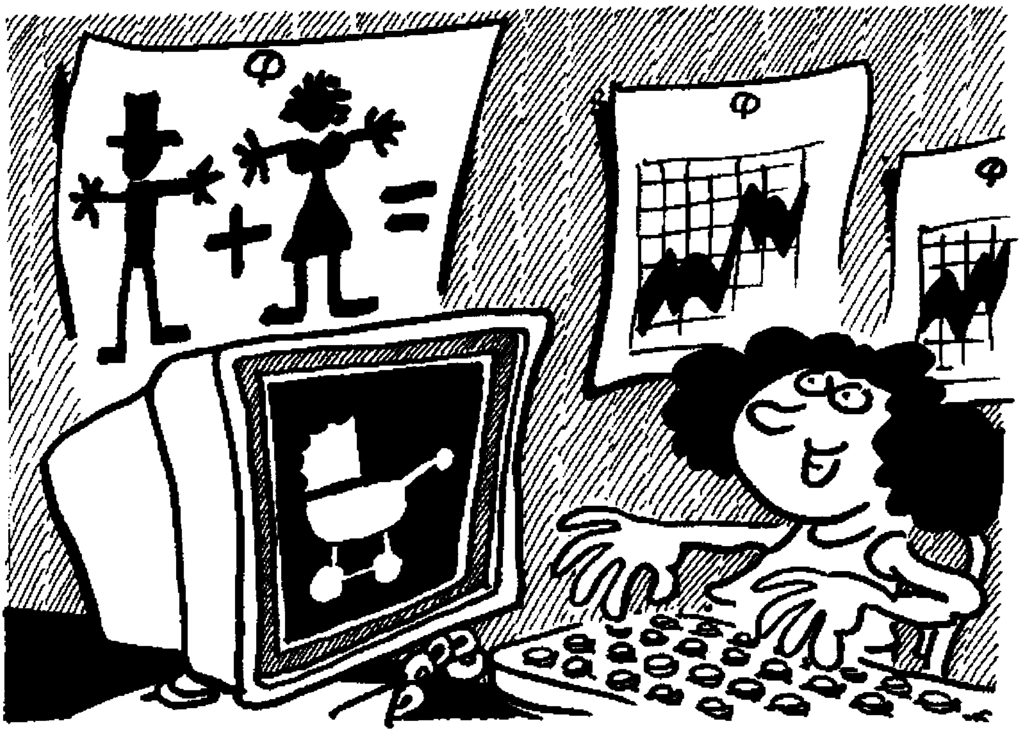
Optional extra resources: Printer Training video or program.

Links

Many statistical packages allow import of data from spreadsheets, and export of data to graphics packages.

Methodology

Students who have once made an involved statistical calculation manually, will always want to use a computer program for this purpose in the future.



Students must acquire the ability to solve routine everyday problems in an algorithmic form (pages 13, 73)

Optional Programming Module

UNIT P1—INTRODUCTION TO PROGRAMMING

Overall Objective of the Unit

Students should be able to design, program and evaluate simple algorithms for elementary task oriented problems. (The term "algorithm" to be interpreted in its broadest sense.)

Context

Programming at this level is not a technical subject; by and large it means changing a task you can "do for your yourself' into one which can be "done by others". This means describing the task as a procedure in sufficient and complete detail so that it can be performed precisely and repetitively by another person or a device.

This "automation" of a task is that what is called programming, sometimes algorithmic programming— one of the main features of informatics. An algorithmic way of thinking and problem solving is needed when using complex or sophisticated tools such as a spreadsheet, a database management system, an operating system or even a word-processor.

Programming is a rapid, specific and suitable way for students to gain experience of solving problems. The objective is not to train "mini-programmers", but to bring students into contact with the programmer's way of thinking. With this aim in mind, the syntactical and other specific features of a programming language are of no importance and have only to be learned on a "need-to-know" basis. Teachers should stress the methodological aspects of problem solving; programming is just a means to an end rather than a goal in itself.

Summary of Content of Unit

The unit contains three main activities: designing a task-oriented algorithm (algorithmic problem solving), translating the design into a program (programming), and bringing the program to life (implementation). These three activities will always be successive and closely tied together. Although the following descriptions treat each activity separately, the order of presentation does not necessarily dictate the order of teaching.

1. DESIGNING A TASK-ORIENTED ALGORITHM (ALGORITHMIC PROBLEM SOLVING)

Objectives

Students should be able to:
1. describe and specify the task to be realised
2. develop an effective and efficient algorithm which realises the identified task, applying a simple, given standard method.

Content

Students have to experiment and identify the steps involved in the process of problem solving, applying simple problem solving strategies in a given format to specify solutions to simple problems drawn from their everyday lives. Examples of suitable problems are: simple simulations of dice throwing, tossing of coins, calculations such as the formulas for areas and volumes or the interest on money loans and deposits, turtle movements in a plane.

Examples of possible programming environments are: Turtle-logo, Basic, Pascal, or an environment for controlling a physical device (e.g. a robot).
When students have built up some expertise, they should learn to subdivide the tasks to be automated into major sub-tasks and into fundamental tasks which are given suitable, meaningful names.

2. TRANSLATING THE DESIGN INTO A PROGRAM (PROGRAMMING)

Objectives

Students should be able to:
1. transform their simple algorithms into computer programs in a procedural language;
2. produce a readable, understandable and user-interactive program.

Content

Teachers should choose simple problems which only require students to use elementary input, output, and assignment operations of a programming language. Where applicable, students should determine proper types and uses of variables; and select suitable and meaningful variable names which represent the functions of the variables in the program.

Students should not be forced to use a separate editor or text processor for entering their program.

3. BRINGING THE PROGRAM TO LIFE (IMPLEMENTATION)

Objectives

Students should be able to:
1. use a given programming environment to enter, edit, compile, debug, update and run the constructed programs;
2. give a meaningful and useful written description of the internal and external behaviour of their program.

Content

Students should store and retrieve programs on and from disks, for further use and modification. They should learn to differentiate between syntax errors and execution errors, and identify possible causes for each type of error. Students should test their programs with given or created test data to determine correctness and limitations and should learn how to obtain printed output and a hard copy of source code.

Teachers should ensure that clear and adequate documentation is produced of all programs so that other users are able to understand the operation of the programs and make further modifications.

Resources

Minimum necessary resources:
One computer between two students (one may plan, while the other enters code or tests);
A simple programming language, preferably with a graphics interface, such as Logo.

Links

This Unit, together with Unit P2, are essential preparation for the General and Vocational Modules at the Advanced Level.

UNIT P2—TOP-DOWN PROGRAM DESIGN

Overall Objective of the Unit

Students should be able to design, program, and evaluate structured algorithms for problems which need a top-down approach.

Context

After first experiences with simple algorithmic design and programming in Unit P1: Introduction to Programming, students should learn to use a top-down approach, while using the same design tools and programming environment. When problems become more complex, the need for a top-down approach more evident. Top-down programming is the normal way in which commercial software is designed and produced.

Summary of Content of Unit

The Unit contains three main activities: top-down design of algorithms, translating the design into a program, and bringing the program to life. As with Unit P1: Introduction to Programming, these three activities will always be successive and closely tied together. Although the following descriptions treat each activity separately, the order of presentation does not necessarily dictate the order of teaching.

1. TOP-DOWN DESIGN OF ALGORITHMS

Objectives

Students should be able to:
1. describe and specify the problem to solve;
2. develop effective and efficient algorithms for the solution of a problem, using a top-down approach through step-wise refinement.

Content

At first the teacher sets some complex problems to be solved, but in the context of ready-made procedures already built and available in the system (and corresponding to parts or sub-problems of the original problem). In a second stage, these procedures will be analysed by students and they will now also construct these procedures themselves.

Some problems should address the processing of strings, using the tools offered by the available programming language. Algorithms of this type are important because they make it evident that only formal processing is to be expected from a computer. Examples of suitable problems are: concatenation, counting the number of words in a text, writing a text in reverse, changing all the vowels in a text, searching for a word in a text.

It is important to point out problems which fall outside the scope of these formal possibilities, such as: translating a text, correcting a text or summarizing a text.

2. TRANSLATING THE DESIGN INTO A PROGRAM

Objectives

Students should be able to:

1. understand that a computer processes "formal objects" only;
2. transform their algorithms into computer programs in a procedural language using formal procedures or functions;
3. produce a procedure-structured, readable, understandable and user-interactive program.

Content

It is important that the top-down designed algorithms are transformed into programs which have procedures and functions, so that students appreciate the need for these constructs.

Before the actual program is written, students should evaluate and debug, step by step, a given, subdivided algorithm; they should create their own set of test data and explore the different parts of the algorithm, but also the algorithm as a whole.

Students should use the following programming constructs: procedure, function, global and local variables, parameters. They should be offered procedures and functions as tools for string processing which they will use to solve text processing problems.

3. BRINGING THE PROGRAM TO LIFE

Objectives

Students should be able to:
1. use a standard programming environment to edit, compile, debug, update and run the constructed programs;
2. give a meaningful and useful written description of the internal and external working of the programs.

Content

Students should store and retrieve programs on and from disks, for further use and modification. They should learn to differentiate between syntax errors and execution errors; and identify possible causes for each type of error.

Students should test their programs with given or created test data to determine correctness and limitations and should learn how to obtain printed output and a hard copy of source code.

Teachers should ensure that clear and adequate documentation is produced of all programs so that other users are able to understand the operation of the programs and make further modifications.

Resources

Minimum necessary resources: One computer per student; suitable programming language or tool.

Optional extra resources: Printer; Liquid crystal display panel for overhead projector.

Links

This unit is essential preparation for General and Vocational Modules at the Advanced Level.

Appendix 2—Informatics in other Disciplines

Informatics can be of considerable value in the teaching of many subjects of the normal curriculum at the foundation and advanced levels. This section gives examples which teachers will wish to use themselves or when promoting informatics more generally with other teaching colleagues. Students will also find these examples a stimulant to their own work in other subjects as well as an enrichment of their informatics studies.

Teachers may find that by integrating the use of computers within subject areas, most of the Computer Literacy objectives can be met without the need for a separate course.

Students of the Advanced Level Unit GA3: Applications of Modelling, will find that earlier experience of using computers in other disciplines provides a good background to their work, as well as good starting points for more advanced subject oriented modelling techniques.

NATURAL SCIENCES

Text Processing
Students can use a word-processor to type their reports of experiments and research.

Desk Top Publishing (DTP)
Students can use DTP to produce attractive looking documents, especially those requiring a combination of text and graphics.

Graphics
Students can use graphics software to prepare illustrations, with or without labels, which can later be imported into DTP documents; or which can replace the usual method of preparing hand-drawings.

Spreadsheets
Students can use spreadsheets to tabulate and calculate results of experiments; or for manipulation of variables to see certain effects that can be more clearly and quickly demonstrated with the use of spreadsheets. Students can also request various types of charts to be plotted from values entered into a spreadsheet.

Teachers could prepare templates to assist students new to spreadsheets; or prepare spreadsheets which have values already entered in order to illustrate effects from the manipulation of variables which is appropriate for work on simulation and modelling.

Databases
Students can create databases such as the characteristics of chemical elements in the periodic table, characteristics of plants, insects, and mammals; and interrogate these databases to find relationships and commonalities. As a first step, teachers could prepare databases into which students can add data.

Robots and Feedback Devices
Students can build robots and use robotics to perform experiments, particularly in physics.
Using mechanical, temperature and other probes to monitor experiments, and feeding the readings directly into a spreadsheet, helps to obtain more reliable results and makes classroom work more realistic. Several software tools exist which take readings, interpret them and present them graphically.

Communication
Students can use computers to communicate with other students on a local network, or with students in other schools both locally and overseas. This enables data to be gathered and shared with others, for example rainfall and Ph values in different countries or particulars of insects unique to a region.

Speech Recognition
In Physics and at times in Biology, students can use external devices to record sounds, and use computer analysis to study sound waves and patterns .

Expert Systems
Expert systems written by students may be of little value but students learn so much from writing such a system that it should be attempted wherever resources are available. One expert system well within the scope of advanced students predicts the result when two chemical elements are mixed.

Modelling and Simulation
The "Three-mile Island" disaster can be simulated in every classroom without any danger to students. Even when students have performed or withnessed a demonstration, repeating the experience through modelling often gives them further insights.

Presentation Software
Students can use presentation software to generate slide presentations of their projects, experiments and research findings to large groups in class.

MATHEMATICS

Spreadsheets
From doing repetitive calculations to showing patterns in certain number manipulations, spreadsheets can play an important role in mathematics at most levels.

Graphics
Specialised graphics packages are available which show the graphical representation of any given function. Also there are software packages which allow geometry problems to be presented on video screen.

Statistics
Using suitable software tools takes the pain of hours of calculation out of statistics and provides important analyses. The complications of manual calculation often make real-world examples too difficult to handle; with a computer, realistic situations can be analysed more readily.

Computer Assisted Design (CAD)
Some CAD packages can be used in some aspects of geometry as substitutes for graphics packages.

Modelling and Simulation
Students can use simple modelling packages, such as Mathematica, to gain insight into mathematical functions.

LANGUAGES

Text Processing
The most common use is to create letters and others documents.
Teachers or students can create Close reading texts, texts with missing words and texts that need punctuation, plural or another tense. Arranging a story in chronological order, or completing a story or an outline are other valuable applications.

Desk Top Publishing (DTP)
Apart from creating newsletters, newspapers and posters, students are extremely fond of using available graphics to create attractive documents.

Graphics
Being able to illustrate what they produce gives students an impetus to write. Students will appreciate ready-to-use graphics, and the high quality graphics they can create themselves. Some programs combine the power of a simple DTP package with sophisticated graphics.

Robotics
For students the control of a robot through commands in a foreign tongue is often a most satisfying task even with a limited vocabulary. For example, Logo is available in English, French, Spanish, Greek, German and a number of other languages.

Communication
Nothing seems to be more motivating for some students as communicating with a native speaker of a foreign language in a distant country. The availability of e-mail, bulletin boards and computer conferencing has made instant communication possible. However, communication by (posted) disk and word-processor document should not be ignored.

Speech Recognition and Synthesis
Given the right software, students can compare their own pronunciation with that of the synthesised model, both oral and visual.

Expert Systems
Given the correct tools, students can write programs which use the rules of an expert system language or manipulate the language. For example, a simple expert system could he written to change English nouns from singular to plural.

Programming
Given a suitable language (Logo, Boxer, Lisp, Smalltalk) students can easily write their own expert systems as indicated above.

"OUTSIDE BACK COVER" —AN ENCOURAGEMENT TO READING

This is an example of an inter-disciplinary project which helps to meet some of the Computer Literacy objectives and links students and teachers in school with information scientists and librarians.

The Project
Students write a "fourth outside back cover" which is a summary of a book recently read with the purpose of giving others the longing to read it (this is not a matter of marketing!).

Typing the Summary
Students use a word-processor to prepare the text, adding personal information about themselves, their class and their school, as well as keywords, ISBN number and an abstract which could be used by school or local libraries. A complementary activity could be a book of the month selection.

Database Creation
Students collect contributions from other students in the same or different classes or schools within their region to create a database of reviewed books.

Database Use
Students interrogate the database for their next reading choice, to get or order a book from the librarian, or to link with the database of the school documentation center.

SOCIAL SCIENCES

Text Processing
Any subject requiring reports, essays and other documents can make good use of a word-processor, a graphics package and desk top publishing software. Advanced students will be able to draw on variety of information sources to prepare multi-media presentations.

Spreadsheets and Databases
In the study of social sciences, spreadsheets and databases serve the same purpose: to enable students to systematise and organise information. For example, students could make use a spreadsheet to make a list of dates, events, countries and persons involved. This list could then be organised by date, by country or by the person's name. Such lists make good study aids. Younger students like to collect information, and will enjoy setting up a database, for example on facts about all EEC countries.

Communication
Students can use e-mail and disk mail to communicate with residents of places of historical or geographic significance.

Statistics
Especially when studying Geography at an advanced level, students may need to use a statistical package.

Expert Systems
Students of Geography could write expert systems on such mailers as the conditions needed for a village to be established and to flourish as a regional trade center.

ART

Graphics
Some graphics packages allow for the creation of original artwork. However, art teachers are more interested in the way in which they can create patterns, complementary patterns and patterns with variety. For example in textile design, computers enable students to see an overall result with less effort than by any other method.

Desk Top Publishing (DTP)
In the design of posters and other printed matter, using DTP ensures a professional product in minimum time, with the option to re-use or modify a design at will.

Programming
Programming languages with a graphics interface, such as Logo, allow for the creation of intricate patterns with minimum of effort. The graphics part of such a language should be included in design courses.

MUSIC—GENERAL OPTION UNIT OP10 GIVES A FULLTREATMENT OF THIS DISCIPLINE.

Appendix 3—General Advanced Level Units

This Advanced Level Module is specified for secondary schools which have the necessary hardware and software, as well as trained teachers able to deliver the units.

Ideally, courses should be built up from Units GA1-3, in consultation with universities and tertiary institutions, so that advanced credit can be obtained towards a tertiary computer science course.

Objective

Students should be able to design and implement technical computer-based systems which model real problems using an algorithmic, problem solving approach.

Context

Units GAS and GA2 together represent a standard set of courses which are commonly used at both secondary and post-secondary level as foundation in a study of Informatics. The depth and breadth of coverage and practice must be adjusted to match the student population so that students will be able to enter Higher Education with basic knowledge and skills in the programming of systems and in software development.

Students who have acquired these skills will often apply them in the modelling and programming of complex problems in areas such as robotics, artificial intelligence, graphics, and mathematics. Unit GA3 provides the comprehensive grounding for such modelling skills.

General Advanced Module

UNIT GA1—FOUNDATIONS OF PROGRAMMING AND SOFTWARE DEVELOPMENT

Objective

Students should be able to design and program relatively small computer-based systems which model process-oriented problems.

Sub-objectives

Students should be able to:

1. adhere to the basic principles of software engineering
2. methodically analyse procedure-oriented problems through decomposition into procedural steps; 3. apply more or less formal techniques to design effective algorithms and data structures;
4. code and realise programs and sub-programs (modules) using a general purpose programming environment
5. use a functional model of a computer system and its programming environment.

Context

Students need to develop skills which are fundamental to informatics as a discipline and to software development at all levels. These include the use of more or less formal methods of problem analysis and program design with an emphasis on creating not too complex programs which meet given specifications and which contain usable user interfaces.

Specific attention is paid to the type of problems which can programmed using fundamental, simple and structured data types (character, integer, real, array, etc.) and basic control structures (sequence, selection and repetition) of a high-level, block-structured programming language.

Emphasis is on modelling through process abstraction (top-down decomposition into sub-processes and subprograms). Methods of analysis, design strategies and the programming language environment should support this type of modelling.

Content

Students will solve several, increasingly-complex problems. Problems are drawn from real application areas which students can readily understand and model. Small individual problems are chosen to learn how to use new algorithmic operations in conjunction with elements already known. But, also, some allencompassing problems should be included in which students have to integrate all that they have learned before.

Problem Analysis

Students apply informal and elementary formal techniques to analyze simple process-oriented problems in many application areas and describe them in terms of sub-process steps.

Design

ALGORITHMS: Students design modular solutions through top-down analysis and stepwise refinement. They choose and specify data structures and algorithms to match the design. The data structures and control structures used in the algorithms should be directly related to the primitive structures of a block-structured programming language.

USER INTERFACE: Students design a simple user interface for their algorithms in the form of a tree of screen designs.

Programming
The algorithms and the modular structure are coded into a general-purpose programming language.

Realisation and Evaluation
The code is then realised in the form available programming environment. limitations of the programs.

Topics

Software Engineering
problem solving process, software life cycle

Analysis
process, input and output specification
identification of steps and modules
informal specification of pre- and post- conditions

Design
top down, modular stepwise refinement
simple, useful user interfaces

Algorithms
simple and nested control structures
simple data structures code structure, readable and useful form
elementary sorting and searching algorithms
simple recursion
design of test data

Realisation
execution, testing, and debugging
documentation
bottom-up testing
incremental realisation

Evaluation
informal comparison of algorithms
limitations of design and program

Programming Environment
hardware components
system software and compilers
representation of stored data

Programming Language Elements
simple and structured data types of language, user-defined types
evaluation of expressions and standard elementary library functions
sequence, control, and iteration structures
simple interactive and text file input and output
subprograms and parameters
local and global variables and scope of variables and subprograms

Resources

Minimum necessary resources:
Version of a high-level block-structured programming language which supports modular program design and contains the necessary data types and control structures.

Introductory level textbook which presents examples of applications using the programming language. Most relatively recent texts, on an introduction to informatics or on computer science organizational-patterns, which have been developed through practice and fit the objectives of this unit

Optional extra resources:
Additional text material on machine organisation, systems software, compilers and language translators, internal representation of information, software engineering and the software life cycle.

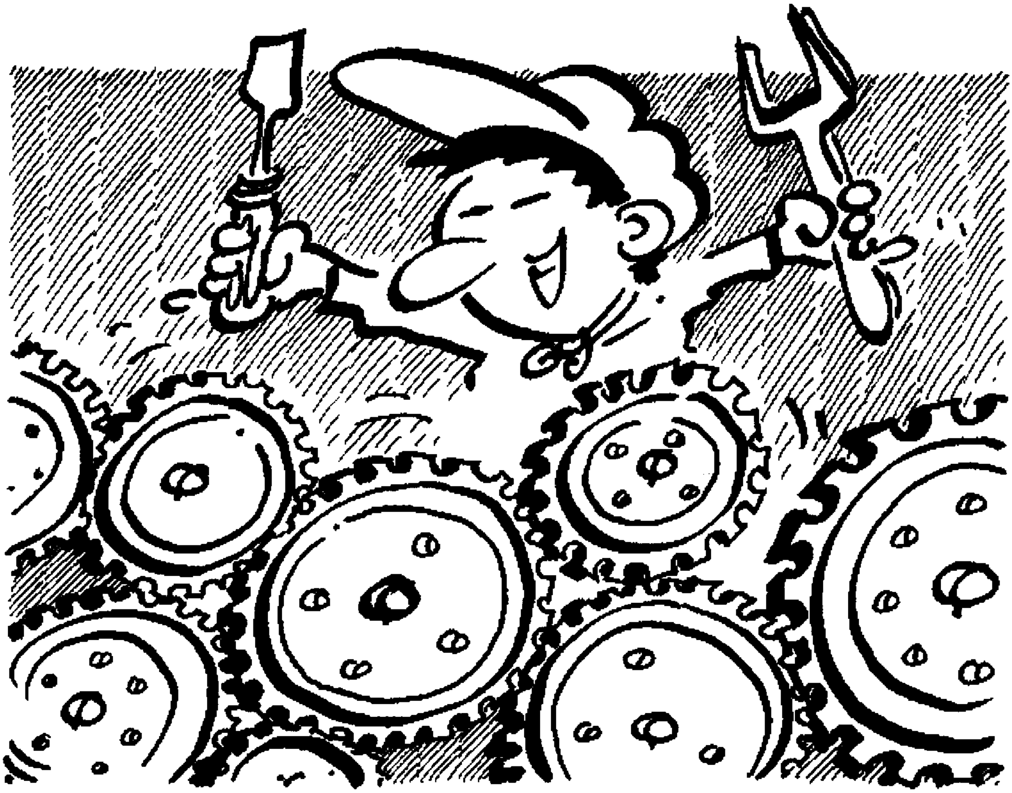
Links

Introduction to Programming (P1); Top-Down Programming (P2); Business Information Systems (VA1).

Methodology

The concepts and skills included in this unit are those which have usually been presented in a first course on informatics for advanced students which some countries will find more appropriate for tertiary institutions than for secondary schools. The unit is quite extensive and requires teaching time every week, extending from over half a year to more than one year The time needed depends on prior experience and student background: whether they have studied Computer Literacy or Programming at Foundation Level. Depending on the target group, a subdivision of the unit may be advisable, organised around the concepts or principles under study.

Emphasis in the unit must be placed on hands-on applications. Students must apply techniques and principles, starting with elementary problems and building up their ability to handle more complex problems. Skills and concepts learned in earlier lessons are constantly reinforced through practice in new problems involving new concepts. Periodic exercises and projects which allow students to synthesize and integrate what they have learned must also be included.



Different countries will be at different phases of development:
Automation Phase (pages 7, 17, 19)

UNIT GA2—ADVANCED ELEMENTS OF PROGRAMMING

Objective

Students should be able to design, program and evaluate relatively complex computer-based systems which model process-oriented problems in many subject and application areas.

Sub-objectives

Students should be able to:

1. methodically analyse and model relatively complex process-oriented problems in a variety of application areas
2. apply moderately advanced formal analysis, design and data abstraction techniques to design effective algorithms, abstract data types and relatively sophisticated data structures;
3. code and realise programs and sub-programs (modules) using a general purpose programming environment
4. evaluate and explore alternative designs to programs.

Context

Students should develop skills for solving more complex and sophisticated problems in many application areas. Emphasis needs to be placed on modelling through data abstraction (an important technique for improving reliability and refuse). Specific attention should be paid to algorithms and strategies for simulating advanced linear and non-linear data structures for the implementation of abstract data types.

Content

This unit extends and builds on unit GAS: Foundations of Programming and Software Development. Students will solve several, increasingly-complex problems from real application areas.

Problem Analysis
Students develop models for relatively complex process-oriented systems using design strategies, such as modelling through abstract data types. They analyse systems to determine basic data objects, and associated functions which are used on these objects in the system.

Design
Students develop abstract data types for the identified objects, which may be reused in other designs which involve the same objects. Students design a solution for the problem expressed in terms of modules which manipulate the abstract data objects only through the identified functions. Students design a suitable and effective user interface.

Programming
Students construct sophisticated linear and tree-like data structures to represent the abstract data types and also construct the functions needed to access these abstract data types in a general purpose programming language which supports information hiding and encapsulation, either directly or through simulation of data abstractions. Students code their design in the programming language.

Realisation
The coded programs are realised and run in the programming environment. Students first realise, test and verify the realisation of the abstract data types, and only then the entire program.

Evaluation
Students determine order of magnitude indicators to compare algorithms, and practice some basic formal methods of program verification.

Alternative approaches
Where time is available, design strategies using other development paradigms, such as object-oriented methodology or logic programming, may be explored if resources make this possible.

Topics

Software engineering
Design for reliability, reuse

Analysis
Analysis strategies, such as data flow analysis using pre- and post-conditions
Formal program verification, assertions, invariants

Design
Data abstraction and information hiding
Effective user interfaces

Algorithms
Graphs and graph algorithms
Encapsulation of abstract data types
Dynamic data types and structures
Binary trees
Binary search trees
Advanced searching algorithms
Efficient non-quadratic sorting algorithms
Hashing methods

Evaluation
Algorithm analysis for order of magnitude approximation
Limitations of algorithms and unsolvable problems
Limits of numerical representations and simple numerical methods

Programming language elements
Singly and doubly linked list representations
Stacks
Queues
Non-linear table data structures

Optional
Applications in graphics, robotics, or artificial intelligence
Modelling using logic or functional programming
Object-oriented programming
Parallel processing of algorithms

Resources

Minimum necessary resources:
A high-level, structured programming language environment, which supports modular program design and data abstraction.
Intermediate level, modern text book on data structures and algorithm analysis which uses a data abstraction approach.

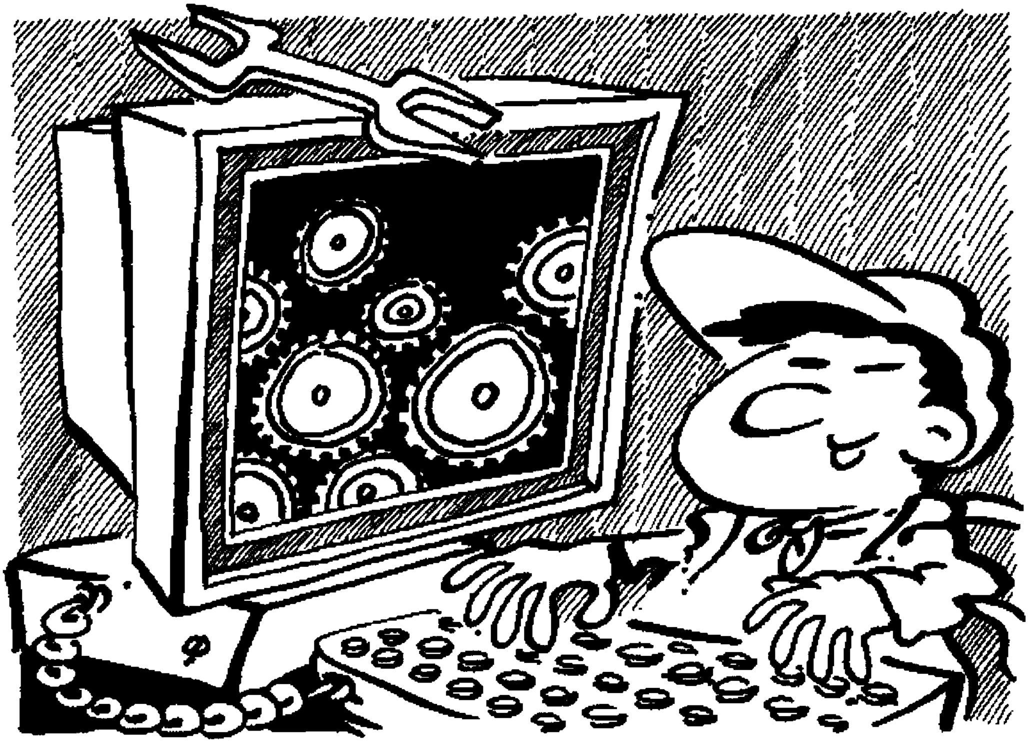
Optional resources
Programming environments for logic, functional, object-oriented programming or parallel processing.

Links

Foundations of Programming and Software Development (GAB)

Methodology

Emphasis is on concepts, theories, and practices of the discipline rather than on exhaustive coverage of language syntax. It is advisable to use, if possible, the same structured programming language for both units. The unit has an established traditional content which is described in many advanced texts on data structures. Teachers and curriculum designers should consider using the structure of these texts as the basis for sub-units and curriculum development.



Different countries will be at different phases of development: Information Phase (pages 7, 17, 19)

UNIT GA3—APPLICATIONS OF MODELLING

Objective

Students should be able to work in teams and specify, methodically model and solve relatively complex problems.

Sub-objectives

Students should be able to integrate previously acquired knowledge, abilities and skills to enable them to:

1. specify, methodically model and solve relatively complex problems with the aid of general purpose programming tools
2. specify, methodically model and solve relatively complex problems in other subject areas with the aid of subject oriented programmable tools;
3. work in teams on a common modelling project.

Context

Modelling ability is fundamental to the successful application of advanced software tools. This Unit requires students to use a variety of computer models, for example simulations and games; to identify the design decisions and simplifications which the developer made; and to discuss their importance and the consequences of possible mistakes.

Students work in teams to find answers to questions from the teacher, involving the creation of a model in which IT concepts, tools and methods play a role. Algorithmic thinking is an important capability for achieving these tasks.

During the development of the project the teacher will coach the students using generalisations, pointing out problems in modelling, team work and project development.
In this way students are able to integrate previously acquired knowledge, abilities and skills gained from their informatics experience and elsewhere.

Content

What is modelling and what tools are available ?
Students work with examples of modelling techniques in informatics and in different subject areas, going through the different phases of a modelling process and making an inventory of classical modelling tools. They study characteristics of team work, examine project management techniques and assess the effects of external pressures on the project. Students are also confronted with examples where modelling implies simplification of reality.

Methodological modelling using appropriate tools—what must be done ?
With a simple question as starting point the description of a problem is first expressed in natural language and then refined by asking appropriate questions and setting limits to the task. Students are then asked to point out components of the "reality to be modelled" which are relevant for constructing the model, and those which can be left out; giving explanations and justifying their choices.
Students write down the precise specifications for the work to be done and make a plan for its realisation. Working together in a team, they search and choose suitable tools and mode fling techniques for solving the problem; they then design and construct the executable solution to the problem.

Evaluation of models and of the process of modelling—how it was done ?
Students first explain and justify their choices of tools, techniques and solutions. They then investigate how closely the solution conforms to the initial description of the problem, updating their original solution as necessary. At this stage they should discuss any social or political issues arising out of the use of incorrect or oversimplified models for real life situations.

Topics for Modelling using General Purpose Programming Tools

Starting points are information modelling techniques such as entity-relationship modelling. Then situations such as growth and decay (for example Forresters model, the Game of Life), the spread of a disease or pollution, and waiting in a queue can be modelled. First an algorithmic model of the situation is made, which is then realised as a general purpose programming language.

Other rich fields for investigation are:

- Elementary Operational Research: linear programming, assignment problems, the travelling salesman problem;
- Probability (tossing coins or rolling dice, lotteries, roulette wheels): for example, a starting point for discussion could be "Everybody knows what is a roulette wheel in a casino. Can you build an "electronic" roulette wheel on your computer and prove that I will have the same chances to succeed or to lose as with a real life roulette wheel ?;
- Graphical problems such as adjacency, paths, loops and shortest path: for example, "Here is a simplified map of the main road of our country; can you represent this map in a diagram on the computer and be able to give the distance when driving from one town to another along one of the roads ?"

Topics for Modelling using Subject Oriented Programming

Here, models can be built and simulated which are part of a particular subject area for which special purpose programming tools exist. For example, problems in Statistics can be modelled and simulated (solved) with a statistical package, models for other mathematical problems can be simulated (solved) with a mathematical tool such as Maple or Mathematics Also more general tools, such as a spreadsheet, can be used to run a simulation.

Appendix 2 gives further examples of modelling opportunities.

Resources

Minimum necessary resources:
General purpose programming language such as Turbo-Pascal;
Software tools such as a data base management system, a text processor, a spreadsheet, depending on the type of problem to be solved, as well as a subject oriented tool such as Mathematica;
Examples of ready made simulations should be available for evaluation and discussion.

Optional extra resources:
Scanners, plotters;
Graphics software, statistics package, Hypertext, Prolog.

Links

Foundations of Programming and Software (GAS), Informatics in Other Disciplines (Appendix 2)

Appendix 4—Vocational Advanced Level Units

This Advanced Level Module is specified for secondary schools which have the necessary hardware and software, as well as trained teachers able to deliver the units.

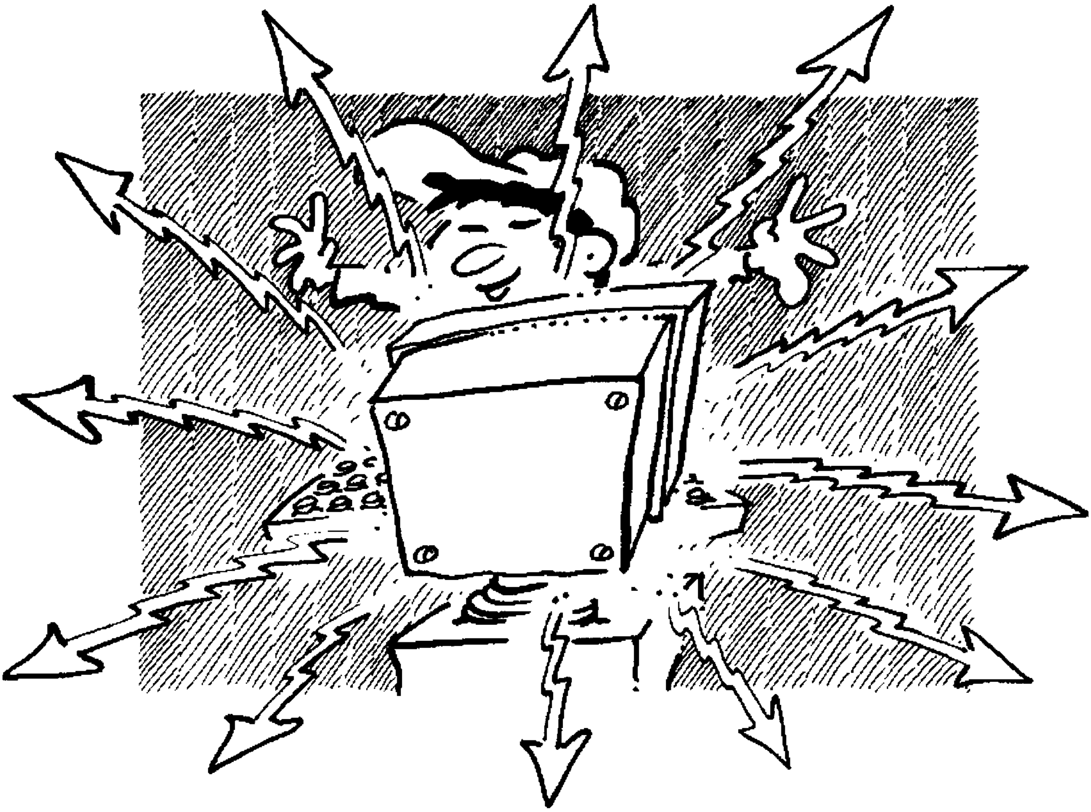
Ideally, courses should be built up from Units VA1-3 in consultation with universities and tertiary institutions so that advanced credit can be obtained towards a tertiary computer science course.

Objectives

Students should be able to:

1. methodically plan, design, realise and implement relatively simple application systems with the aid of problem oriented tools;
2. identify problems involved in project management.

Instead of studying Units VAN, VA2 and VA3 in sequence, Unit VA3 may be used to provide a realistic context within which the objectives of Units VAN and VA2 can be met.



Different countries will be at different phases of development
Communication Phase (pages 7, 18, 19)

Vocational Advanced Module

UNIT VA1—BUSINESS INFORMATION SYSTEMS

This unit assumes basic competence in the use of computer systems and programming skills as developed in Units Pa and P2.

Overall Objective of the Unit

Students should be able methodically to plan, design, realise and implement relatively simple information systems with the aid of problem oriented tools.

1. INFORMATION PLANNING

Objective

Students should be able to use formal methods of data flow analysis to identify requirements for an existing information system.

Context

Students should be able to relate programming to the real world, and especially the world in which they are likely to find employment. The relevance of the material will enhance the value of the course and make it more interesting for both student and teacher.

Content

Students will analyse a simple case study of, for example, a ware house or an accounting office which is described in such a way as to provide the necessary information. The case study will be so constructed that no advanced use of techniques is required. The techniques used will be based on diagrams.

Students will work in groups using different methods to collect data such as interviews and questionnaires. The forms and other documents used in the case-study organisation will be analysed as sources of accurate information.

The following student activities are part of Information Planning phase of the unit:
a) sketch the data flow within a given organisation
b) quantify the volume of data flowing in the organisation;
c) assess the speed of data flow within the organisation;
d) specify a relational data model;
e) identify functions to be provided to employees by the Formation system.

2. DESIGN

Objective

Students will be able to develop a design of a simple database and a limited number of associated functions for data entry, data retrieval, data updating and data presentation.

Context

The case study should only require a few functions to be specified and the required human-computer interface will be of a simple, standard form. The teacher could play the collective role of the users.

Content

In designing a database students will need to be accurate by specifying the overall structure and the specific details for the variables in relation to the data to be used. It will be necessary to review specifications with users to ensure that the specifications are comprehensive and appropriate.

The following student activities are part of the Design Phase of this unit: a) specify the structure of the database; b) specify sample data to be included in the database; c) specify the necessary functions and the human-computer interface; d) design a small program in terms of the database, the functions and the interface.

3. REALISATION

Objective

Students should be able to realise and test the designed program on a computer.

Context

Students will typically use a programmable database system like dBase or a structured, general purpose programming language with a comprehensive library of pre-programmed database functions.

Content

Before trial-running the program with test data, students will need to establish criteria to be used in the evaluation, testing of specifications and coding. Proper documentation will be necessary to enable students to trace the sources of errors.

The following student activities are part of the Realisation Phase of this unit:
a) transform the designed program into a real program;
b) specify testing criteria for the program;
c) trial-run the program with test data;
d) test and verify the coding;
e) identify problems areas and provide solutions.

4. Implementation into the Organisation

Objective

Students should develop a feeling for the problems associated with the implementation of an information system into an organisation.

Context

The different employee functions in the case-study organisation could be played by students. The teacher could supervise the role playing and help in identifying the problems.

Content

Students will have their design, and specifications of their program validated with respect to the real life situation. They will categorise the nature of the problems into classes: data model, specification, functions and interface. The teacher will bring their attention to the importance of documentation.

The following student activities are part of the Realisation Phase of this unit:
a) implement the information system into the role played organisation;
b) identify problems with the use of the system;
c) suggest practical solutions to these problems.

5. USE IN ORGANISATION

Objective

Students should be able to evaluate the effectiveness of the running of their program within the organisation.

Context

Students will run the system simulating the organisation involved. The technical skills of the students will be put to test by this life-like environment, thus providing students with an opportunity to evaluate their ability to identify the capabilities and limitations of their system.

Content

Using data specifications, students will evaluate the capabilities and limitations of their system. Software and hardware interface errors will have to be corrected, enhancing the design.

The following student activities are part of the Use in Organisation Phase of this unit:
a) monitor the effectiveness and efficiency of the system
b) evaluate the capabilities and limitations of the system to handle the activities in the organisation.

Resources

Documentation from previously developed systems should be made available to students. Although it is likely to be aimed at tertiary students, a relevant book on Systems Development would be helpful as background material.
A programmable database system like dBase or a structured, general purpose programming language with a comprehensive library of pre-programmed database functions should also be available to students.

Links

Working with a Database (C6); Database Design and Use (E1)

UNIT VA2—PROCESS CONTROL SYSTEMS

This unit assumes basic competence in the use of computer systems and programming skills as deve - loped in Units P1 and P2

Objective

Students should be able methodically to plan, design, realise and implement relatively simple process control systems with the aid of problem oriented tools.

1. PLANNING OF PROCESS CONTROL

Objective

Students should be able to identify requirements for technical systems which control or automate processes within some environment.

Context

Students will develop an awareness of the various functions of control systems needed to monitor technical systems within organisations. After that they will work on a case study connected with a simple process to control, such as a robot.

Content

The following student activities are part of the Planning Phase of this unit:
a) use existing control systems to monitor technical systems;
b) specify problems within a given technical environment;
c) identify the need for technical process control within the given environment;
d) identify the input and output data which are needed to control the system under consideration;
e) specify the functions needed to control the system under consideration.

2. SYSTEM DESIGN

Objective

Students should be able to design simple systems which control and monitor technical processes.

Context

In preparation for the world of employment, students should be able to analyse a simple case study connected with a simple device, such as a robot. They should also be able to design and enhance the technical operations of the control system.

Content

Using a case study, students will develop techniques to write a program to support an existing system for efficient operations. The case study will be limited to the use of simple sequential procedural algorithms.

The following student activities are part of the System Design Phase of this unit:
a) produce technical specifications of the procedures needed
b) design the procedures of the controlling system.

3. Realisation

Objective

Students should be able to program the simple procedures for process control.

Context

Students should be able to translate the designed procedures into a program for a problem oriented programming environment (for example a language for robot control) or for a general purpose programming language with a comprehensive library of pre-programmed control procedures.

Content

The following student activities are part of the Realisation Phase of this unit:
a) realise small programs designed to support control operations;
b) specify testing criteria for the program;
c) trial-run the program for testing purposes;
d) test and verify the coding;
e) identify problems areas and provide solutions.

4. Implementation into the Environment

Objective

Students should develop an appreciation for the problems associated with implementation of a control system into the (organisational or technical) environment.

Sub-objectives

Students should be able to:
a) identify any technical problems associated with the implementation of the control system program;
b) classify any problems relating to the software and hardware interfaces.

Context

Using a real life, case-study environment the students should be able to evaluate and validate the specification, design and coding of their program. For example, the controlled robot could be part of a production line. The operation of the production line can be simulated by the students.

Content

Students will have their design and specifications of their program validated with respect to a real life situation. They will categorise the nature of any problems into classes: data, specification, control functions and interface. The teacher will bring their attention to the importance of documentation.

The following student activities are part of the Implementation phase of this unit:
a) implement the controlled system into the role played environment;
b) identify problems with the use of the system;
c) suggest practical solutions to these problems.

UNIT VA3—PROJECT MANAGEMENT

Objective

Students should be aware of the main variables which influence the progress and success of a project and are able to plan team activities within a given, but not too long, time frame.

Sub-objectives

Students should be able to:
1. identify the goals of a project, all the variables (social, political, financial, economic, cultural and human resources) which are likely to affect the implementation of the project;
2. plan team activities and use a simple graphical planning tool.

1. PLANNING PHASE OF A PROJECT

Objective

Students should be able to Identify the objectives of the project and have an appreciation of the variables which may affect successful implementation.

Context

Team work is essential in the modern information technology environment. Students should work on a project as a team in the Planning Phase (see also Units V.A. 1 and V.A.2). It should be ensured that students understand and appreciate their respective roles as members of the team.

Content

The objectives and parameters of the project have to be defined to ensure that all the variables will be considered at the appropriate stages of the project. Students will be made aware that market research should be undertaken to ensure that decisions affecting the project will be based on quality information.

Sub-objectives

Students should be able to, as part of the planning process:
a) identify the objectives of the project to be implemented;
b) state the objectives in operational terms
c) identify the social, cultural, political, financial and human resources variables which may influence the project.

2. DESIGN PHASE AND REALISATION PHASE OFTHE PROJECT

Objective

Students should be able to plan team activities and use a simple graphical planning tool.

Context

In the real world, having the best plans means nothing if you cannot "sell" them within the organisation. Students should give a mock presentation of their project in their respective teams while another team evaluates, pointing out what they see as problems and asking the team presenting to justify their choices and strategies of the suggested way forward.

Content

The following student activities are part of the Design Phase and Realisation Phase:
a) identify all the essential output from the project;
b) quantify the output in terms of time, materials, financial and human resources
c) plan the application of resources with a simple graphical planning tool.

3. IMPLEMENTATION PHASE OFTHE PROJECT

Objective

Students should be able to relate the quality of the outcome of a project to the planning and realisation phases, and to the monitoring of the project during these two phases.

Context

In the real world, we learn and make progress by analysing the good and the bad of every project; planning to re-utilise the good and avoid the bad in future. Students should discuss, under the directions of the teacher, how each of the identified variables has affected the implementation of the project and what should be done to minimise negative effects. Each team presents a comprehensive picture of what they propose should be done.

Content

The following student activities are part of the Implementation Phase:
a) collect data relevant to the quality of the implementation of the project;
b) identify the factors which have adversely influenced the project
c) suggest what could have been done to improve effectivity of the project.

Appendix 5—Bibliography

For most units, books can be found in many languages in which the unit is worked out in detail as teaching material. Such books can offer guidance to text book writers implementing the units of this curriculum for their local situation.

More general texts, treating internationally identified problems and solutions associated with Informatics and Information Technology in Secondary Education are regularly produced by the International Federation for Information Processing, Technical Committee 3, Working Group 3.1. The most recent, relevant texts are:

F. Lovis & E.D. Tagg (eds), Computers in Education, proceedings of IFIP European Conference on Computers in Education ECCE 88, North-Holland, 1988.

A. McDougall a C. Cowling (eds.), Computers in Education, proceedings of the IFIPTC3 Fifth World Conference on Computers in Education (WCCE 90), Sydney, Australia, July 1990, North-Holland, 1990.

B. Samways & T.J. van Weert (eds.), Impact of Informatics on the Organisation of Education, Elsevier, 1991.

D.C. Johnson a B. Samways (eds.), Informatics and Changes in Learning, Proceedings of the Open Conference on Informatics and Changes in Learning, Gmunden, Austria, June 1993, Elsevier Science Publishers, 1993.

GOOD PRACTICE GUIDES

The series Guidelines for Good Practice produced by IFIP Working Group 3.1 offers monographs on Informatics and Information Technology in Secondary Education which contain extensive bibliographies. These monographs may be obtained through IFIP Secretariat, 19 Place Longmalle CH-1204, Geneva, Switzerland:

Harriet G Taylor, Robert M. Aiken, Tom J. van Weert, Informatics Education in Secondary Schools, IFIP Working Group 3.1 Guidelines for Good Practice, 1991.

Ferran Ruiz i Terrago, Integration of Information Technology into Secondary Education: Main Issues and Perspectives, IFIP Working Group 3.1 Guidelines for Good Practice,
Tom J . van Weert (ed.), 1993.

Translations and Copyright

This curriculum has been developed to be of use to many countries in many circumstances. Its level of description is such that textbook writers will, on the basis of this curriculum, be able to produce texts suitable for teaching in a local situation. UNESCO may be of help in this by stimulating the conversion of this curriculum into actual teaching texts in local settings.

The curriculum has been written in English. To reach the world community, translations are required into at least: Chinese, French, Russian and Spanish. Care should be taken to involve the right experts on any translation; such translations will have to be authorised by UNESCO.

Layout: Georges-Alain Dupanloup
Illustrations: Pécub