ICT106
Fundamentals of Computer Systems

Unit Outline

Semester 2, 2006

Unit Coordinator:
Associate Professor Lance C.C. Fung

School of Information Technology
Division of Arts

MURDOCH UNIVERSITY
PERTH, WESTERN AUSTRALIA
First Offering: 1999

Unit Coordinators:

1999   Dr Pyara Dhillon
2000   Dr Pyara Dhillon
2001   Dr Pyara Dhillon
2002   Dr Pyara Dhillon
2003   Associate Professor Lance C.C. Fung
2004   Associate Professor Lance C.C. Fung
2005   Associate Professor Lance C.C. Fung
2006   Associate Professor Lance C.C. Fung

Unit Lecturer:       Associate Professor Lance C.C. Fung
Unit Tutors:         To be announced
## Table of Contents

### Study Guide

1. Unit Coordinator & Contacts  
2. Unit Outline  
3. Unit Organisation  
4. Unit Materials and Computing Facilities  
5. Required Course-work and Timetable  
6. Assessment  
7. Assignment Submission Requirements  
8. Teaching Week Table  
9. Advice on How to Study this Unit  

### Topics

<table>
<thead>
<tr>
<th>Topic</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong></td>
<td>Introduction to Computer System architecture and C programming language</td>
</tr>
<tr>
<td><strong>2.</strong></td>
<td>Information Representation and Storage. Data Input and Output in C</td>
</tr>
<tr>
<td><strong>3.</strong></td>
<td>Arrays, functions and structures in C, Using the Debugger</td>
</tr>
<tr>
<td><strong>4.</strong></td>
<td>Introduction to 80x86 Microprocessors. Bit manipulation and Pointers in C</td>
</tr>
<tr>
<td><strong>5.</strong></td>
<td>Machine language and Introduction to the 80x86 Instruction set</td>
</tr>
<tr>
<td><strong>6.</strong></td>
<td>Assembly Language Fundamentals: Generating, Loading and Executing programs</td>
</tr>
<tr>
<td><strong>7.</strong></td>
<td>Assembly Language Programming Development. Addressing modes. Using the assembler</td>
</tr>
<tr>
<td><strong>8.</strong></td>
<td>High-Level Language interface - Inline assembly in C</td>
</tr>
<tr>
<td><strong>9.</strong></td>
<td>Procedures and Functions. Calling and Exit Conventions, Run-time Stack</td>
</tr>
<tr>
<td><strong>10.</strong></td>
<td>HLL Control Constructs and Integer arithmetic at Assembly Level</td>
</tr>
<tr>
<td><strong>11.</strong></td>
<td>Concepts of Input, Output and Interrupts</td>
</tr>
<tr>
<td><strong>12.</strong></td>
<td>Introduction to Operating Systems. Memory Management.</td>
</tr>
</tbody>
</table>
1. Unit Coordinator and Contacts

Unit Coordinator
Associate Professor Lance C.C. Fung, School of Information Technology
Tel: (08) 9360 7507
Office: Transportable 1, Room 1.06
email: L.Fung@murdoch.edu.au

Please do not hesitate to contact the unit coordinator if you have any difficulties and queries about the unit.


The above website contains information from the Handbook, Study Support, Assessment policies and grading.


The unit ftp site will be the repository of unit materials. Definitive unit details, announcements facility and tutor contact are provided. Anything from the topic scheduled or presented in class is eligible for inclusion in the exam, with answers referenced to the textbook and materials developed by this class.

Tutor for External Students who enrolled under the X-Mode
Details about your tutor will be provided to you by the External Studies Office by the end of week 2 of the semester. The tutor will mark your assignments and assist you with your study. Do not hesitate to contact your tutor if you have any questions.

Administrative Contacts
If you have any queries about your enrolment in this unit please contact:
The Division of Arts student administration office:
Tel: (08) 9360 2420
Fax: (08) 9310 6958
Email: arts@murdoch.edu.au
2. Unit Outline

Introduction

This unit builds upon your programming background gained from previous unit such as ICT102, Introduction to Computer Science. The main aim of ICT106 is to provide an introduction to the operation and structure of computer systems. The content of the unit is concerned with programming in C, Assembly language and the low level operations of a computer system. Although this unit is not concerned with the electronic details of a computer, we will study the major functional components of a computer system from the point of view of low level programming (assembly language) and how these low level characteristics relate to various aspects of programming in a high level language (HLL). Additionally, the unit provides an introduction to operating systems by addressing the machine or system level aspects that are essential in supporting a modern operating system.

From a practical point of view, this unit provides an introduction to C and assembly language programming for the IBM PC. However, the general approach and methodology should facilitate, if needed, assembly language programming on any other type of computer. For many students interested in computer science, the need to program in assembly language is of less importance than it has been in the past. However, the concepts and constructs used in a HLL become much clearer in the process of learning about the low level characteristics of a computer. In addition, with the increasing range of dedicated or “embedded” computer-based systems, the need for C and Assembly skills is expected to be in demand by the industry.

In ICT106 the emphasis is on the study of computer as a whole system and how its low level characteristics directly relate to programming and operating systems. Unlike the approach used in many textbooks (including the one for this unit), few people working in the field of computing write large programs entirely in assembly language. Instead, it is much more likely that the bulk of a program is written in a HLL, with only one or two routines (when needed) being written directly in assembly language. Such assembly language programming is for the purposes of either speed enhancement or access to low level features not directly provided by the HLL. In addition, when timing and memory requirements are part of the constraints of a system specification (such as real-time systems or mission-critical systems), assembly language may be the last resort to meet the requirements. It may be possible that in your working life, you will be asked to write application or system programs entirely in an assembly language for a specific computer system. This is usually for small and simple embedded processors or dedicated application routines. The understanding of the computer system that you will gain and the techniques that you will learn in this unit will serve you well.

Unlike the conventional and traditional approaches of linking assembly language routines to those of a HLL, the now common place Integrated Development Environments (IDE) greatly facilitate the ability of writing and using assembly language routines.
This study guide is for use by both internal and external students. For internals students studying on campus, this outline is supplemented with the information presented in lectures and tutorials. It is necessary for both internal and external students to download a copy of the lecture notes as well as other material from the unit's ftp site. Note that assignments and practice work sheets will also be posted at the ftp site. All students should therefore check the ftp site regularly for updates.

Whether you are an internal or external student, please read the following pages carefully for advice on how you should study in this unit, and what material you need to submit for assessment. Most sections of the study guide deal with all students, some sections just with internal students, others just with external students. These are clearly marked, so make sure you are reading the correct sections. This is particularly important for the assessment sections, as there could be some minor differences corresponding to your enrolment option in this unit.

Pre-requisites

ICT102 Introduction to Computer Science

Note. Students must either have passed ICT102 or been formally granted an exemption from it. Exemptions must be approved by the Programme Chair, Dr Pyara Dhillon (Tel: 08-9360 2799; Office: ECL 2.048, email: p.dhillon@murdoch.edu.au)

A working knowledge of a high level language (e.g., Java, C/C++, Pascal) is also assumed for this unit.

Aims & Objectives

The aims of this unit are to provide

- an introduction to the low level machine aspects of a computer system
- a framework with which to appreciate how a computer works at the machine level
- an introduction to programming in C and assembly language
- a basis for understanding how high level languages utilize machine resources
- an introduction to the machine level aspects of an operating system
- an introduction to the peripheral and computer architecture

At the end of this unit you should

- be able to program in the high level language C
- gain an understanding of assembly language programming
- be familiar with the relationships between C and assembly language
- gain some understanding of the organisation of 80X86 based computers
- understand some of the low level aspects of an operating system
- gain some understanding of some common peripherals in general purpose and dedicated computer systems.
3. Unit Organisation

Structure

The unit material is divided into 12 topics and associated practical exercises. For the internal students, there are 3 hours of lectures and a two-hour practice session each week. The practice sessions are used to exercise and reinforce knowledge of the unit material and are a vital part of the learning activities of the unit. External students need to work through the exercises as well. You will also need to spend time on reading and complete the practical work outside the scheduled hours in order to fully understand the unit material. It will help your understanding of the material if you read the relevant text book chapter in conjunction with the notes for each lecture topic.

Here is a proposed approximate schedule of topics:

<table>
<thead>
<tr>
<th>Starting Week</th>
<th>Topic No.</th>
<th>Topic Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Introduction to Computer System architecture and C programming language</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Information Representation and Storage. Data Input and Output</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Basic Data structures and format. Arrays, functions and structures</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Introduction to Microprocessors and computer architecture.</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Machine language and Introduction to the assembly language Instruction set</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Assembly Language Fundamentals: Generating, Loading and Executing programs</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Assembly Language Programming Development. Addressing modes.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>High-Level Language interface. CPU and Memory</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>Procedures and Functions. Calling and Exit Conventions, Run-time Stack.</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>Modern Computer Systems. HLL Control Constructs and Integer arithmetic at Assembly Level</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>File Management and Programming Tools</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>Unit Review</td>
</tr>
</tbody>
</table>
On-campus Attendance:

**External Students:** No on-campus attendance is required. However, you are welcome to attend as many of the on-campus lectures and tutorials as you wish. Please contact the Unit Coordinator or the Programme Secretary for details about the timetable and venues, or visit the web site: http://www.murdoch.edu.au/admin/timetables/teaching/

**Internal students:** are expected to attend three hours of lectures and one 2-hour practice session each week.

Electronic Communication - email, etc.

The university automatically creates e-mail accounts for all students. These accounts are created on the student machine student.murdoch.edu.au. It is the student's responsibility to regularly read the emails sent to their student email account at Murdoch University. If you are using some other email address on a regular basis then it is your responsibility to arrange forwarding of your Murdoch email to this address. The ITS help desk can provide help on how to set up email forwarding.

To locate the online forms for email redirection and other personal details, visit the following site: http://wwwstudent.murdoch.edu.au/services/infosheet/#email

Change of Enrolment

If you are maintaining a steady rate of progress through the unit, then by the end of week 5 you will have a good idea as to your ability to satisfactorily complete the unit. At this stage in the unit, if you have not satisfactorily completed most of the work required for the first three topics, you should discuss your progress with your Tutor.

To make any change of enrolment, submit a Change of Enrolment form to the Division Executive Officer at the Division of Arts. The deadline for such a change is **HECS census date, which for second semester units is August 31.**

Important Deadlines

Students should be aware of the implications of different dates of withdrawal from the unit.

- **Enrolment - end of Week 3:** Does not appear on academic record
- **Week 4 - end of Week 10** Appears as "Withdrawn" on academic record
- **After Week 10:** Appears as "Fail" on academic record.
- **After 31st August in Semester 2:** Incurs HECS liability
### 4. Unit Materials and Computing Facilities

**Materials**
You will be expected to keep up to date with reading

1. the textbook,
2. additional readings given out during the semester
3. lecture notes,
4. weekly practice sheets and
5. other hand-outs which will appear from time to time.

The unit material will be available progressively on the [unit's ftp site](http://ftp.it.murdoch.edu.au/pub/units/ICT106/). Internet access is required to access the unit's ftp site. You can have your own private access or regular access through a friend, neighbour, relative, employer or university. You will also need a web browser like Netscape or Internet Explorer. **You must visit the ftp site regularly at**

**TEXT BOOK**


OR


**Textbook**
Irvine’s book is used as the unit text as it contains all the basic and fundamental topics on assembly language, processor architecture and introduction to operating system including 16-bit DOS and 32-bit Windows. This is a good reference text on the topic if you continue your exploration on the low-level aspects of the computer. However, the 4th Edition is no longer in print and the 5th Edition is supposed to be available in Mid-June. While there are some improvement on 5th Edition, the 4th Edition will be sufficient for the purpose of this unit.

**References**
For reference purposes or for further study some relevant titles are given below:


(This is a useful reference as it also includes C++ and Java. All these languages are used in the course. In addition, the book also includes a CD which contains the Borland C++ 5.5 Compiler for Windows and J Builder)
Foundation for Windows, Linux and Solaris.)


(This book uses the traditional modular approach to C program development. It contains a CD with a special edition of Microsoft Visual C++ compiler.)


**Safari Tech Books Online**

Murdoch Library also has a collection of online IT books. You should check the following link for relevant topics relating to this unit.

http://prospero.murdoch.edu.au/search/t?SEARCH=safari+tech&sortdropdown=-&searchscope=1&submit=Submit

**Computing Facilities - Internal Students**

The computer laboratories used for ICT106 are available during normal opening hours. Arrangements to use computer laboratories outside these hours can be made by seeing the Division Office staff. You will need to bring in a passport size photograph when making an application for a lab access pass.

For access to the building during week-ends, contact the security guard at the Chancellery Building. Check times when a laboratory is reserved for formal practice classes during weekday mornings and afternoons.

**Computing Facilities - External Students**

External students will have to arrange their own access to a computer and the required software. This can be done in several ways.

1. Buy your own.

   **Hardware:** The minimum specification is a Pentium processor running Windows 95, 98 or better, minimum 128 Megabytes of RAM, 500 MB
of hard disk space, floppy disk drive, CD-ROM drive and a printer. A more powerful machine will certainly make the experience more enjoyable. These days, you should be looking for a Pentium 4 based PC with a minimum of 256MB RAM, 20GB hard disk, a CD/DVD-ROM and runs under Windows ME or XP.

**Software:**

Operating System: DOS with Windows 95/98 or better.

An Acrobat Reader is required for all lecture notes.

A web-browser to read documentation and the unit’s web page.

**Compiler:**

Cygwin and GCC will be used to emulate the Linux Operating System and development of C programs.

Other compilers are Microsoft Visual C, Borland Turbo C and Pacific C.

**Assembler:**

Microsoft Macro Assembler (MASM) with linker and debugger. *Note that MASM 6.11 and 6.15, and the Code View debugger are included on the CD that comes with the unit textbook (4th Edition).*

**Editor:** Any one of the following:

1. Notepad
2. Edit
3. any other text editor (must not be a Word Processor unless the file is stored in .TXT instead of .DOC)

2. Use the on-campus facilities at Murdoch. Refer to the information given in the above section for internal students.

**Network:** Internet access is required to access the unit’s ftp site. You can have your own private access through an ISP (Internet Service Provider) or regular access through a friend, neighbour, relative, employer or university. You will also need a web browser like Netscape or Internet Explorer. You must visit the ftp site regularly. [http://ftp.it.murdoch.edu.au/pub/units/ICT106/](http://ftp.it.murdoch.edu.au/pub/units/ICT106/)

**NOTE:** If any of these resources are not available to you then you will not be able to complete the unit and so you should seek advice immediately or withdraw from the unit.
5. **Required Course-work and Timetable**

**Workload** Full-time students usually take 12 points per semester and so, with 3 points credit, this unit represents about one quarter of a full-time load, or about 10 hours per week. If you are studying the unit over two semesters, the weekly workload should be halved.

**Assignments and Regular Practice Work**

*Continuous assessment* for the Unit will consist of a *Project and a number of practice exercises*.

**how to submit?**

External Students: Mail or deliver your assignments to the External Studies Office, and not to your tutor (UNLESS REQUESTED BY THE TUTOR OR WITH ARRANGEMENT WITH THE TUTOR). Assignments may be accepted by fax at (08) 9360-6017. To assist the transmission of clean copy, use black biro or type on white A4 paper and include an Assignment Attachment sheet with sections A, B and C completed. If you have faxed it, do not send the original assignment by mail (except for any disks of course). If you attach a self-stamped Assignment Receipt Card to your assignment, the Assignment Clerks will date stamp it and return it to you by mail.

**keep a copy** All Students: Be sure to keep a copy of all assignments submitted for assessment in case they get lost in the mail.

**Assignment Timetable**

It is in your best interest to work steadily in this unit, rather than to leave much of the work to the last minute. You must aware that most units may set similar deadlines for assignment submission during the semester. Hence, being too many assignments to be submitted within the same week is not an excuse as you have been informed well ahead. Also, if you submit your work regularly, then you will receive feedback from your tutor at times when it is most useful (and not after you’ve taken the final examination, for example).

**Due Dates:**

The assignment due dates are:

<table>
<thead>
<tr>
<th>Assessment</th>
<th>% mark</th>
<th>Week Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Exercise 1</td>
<td>6%</td>
<td>Week 5</td>
</tr>
<tr>
<td>Project Abstract</td>
<td>5%</td>
<td>Week 5</td>
</tr>
<tr>
<td>C Exercise 2</td>
<td>8%</td>
<td>Week 9</td>
</tr>
<tr>
<td>C Project</td>
<td>15%</td>
<td>Week 11</td>
</tr>
<tr>
<td>Assembly Exercise</td>
<td>6%</td>
<td>Week 12</td>
</tr>
<tr>
<td>Assembly Program</td>
<td>10%</td>
<td>Week 13</td>
</tr>
</tbody>
</table>
The actual due dates will be specified on the assignments themselves. The assignments will be made available several weeks before they are due. Check unit's ftp site and announcements on the unit web page.

**Late submissions will incur a 10% deduction per day late, based upon the total possible mark for the assignment.**

**External students must also submit their solutions to the exercises.**

**Internal students are required to demonstrate their exercises during the tutorials.**

Please discuss with your tutor if you plan to deviate significantly from a due date at any time during the semester.

**External Students**

External students should aim to post or deliver assignments to arrive by the due date.

**Having difficulty?**

If you cannot complete all the questions in an assignment, you may wish to send in what you have completed by the due date. If you need an extension for an assignment, contact the unit coordinator before the due date has passed. **Extensions are granted only in exceptional circumstances.** In requesting an extension you should also suggest for consideration a schedule for submission which meets your circumstances. Alternative work may be required if information about the solution has been distributed or assignments returned.

Unless you have obtained a prior extension to the submission deadline, be advised that late submissions will incur a penalty of 10% deduction per day late, based upon the total possible mark for the assignment.

**Final submission date**

Assignment submissions cannot be accepted after the examination has commenced unless a written application for deferred assessment has been lodged with and approved by the Office of Student Services in accordance with the deferred assessment procedures given in Degree Regulation 46.

**Assignment clerks**

Contact the Assignment Clerk if you have queries about the return of your assignments.

Tel: (08) 9360-2708
Fax: (08) 9360-6017
Email: assignments@murdoch.edu.au
6. Assessment

Contact your lecturer if you require further explanation of the grades allocated for your assignments. Students may inspect their marked examination scripts and discuss the marking with the Course Coordinator within 14 days of the posting of results (Degree Regulation 43). Appeals against the final grade must be submitted in writing to the Dean according to the procedures given in Degree Regulation 49. For further details about assessment see the current University Handbook and Calendar, in particular Degree Regulations 40-51.

Types of Assessment

This unit will be assessed by regular practice work, assignments, and a final examination with the following weights:

<table>
<thead>
<tr>
<th>Continuous Assessment:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C Exercises (x2)</td>
<td>14%</td>
</tr>
<tr>
<td>C Project (Abstract, Report and Program)</td>
<td>20%</td>
</tr>
<tr>
<td>Assembly Language Exercise</td>
<td>6%</td>
</tr>
<tr>
<td>Assembly Language Program</td>
<td>10%</td>
</tr>
<tr>
<td><strong>Continuous Assessment</strong></td>
<td><strong>50%</strong></td>
</tr>
</tbody>
</table>

| Final Examination (Closed-book)         | 50%|

Final Examination

At the end of the unit a three-hour examination will be held, based on all the material of the unit.

**when?**
The examination will be held in November.

**where?**
The University Examinations Officer will contact external students about the time and venue for the final examination. Perth metropolitan students are expected to take the examination on campus, while arrangements will be made for non-metropolitan students to take the final examination locally.

**memory aids**
In the examination, **you will not be permitted to take in any calculators. However, you are permitted to bring in ONE PAGE (DOUBLE-SIDED) HAND-WRITTEN NOTES.**

**identification**
The University requires that all students sitting end-of-semester examinations (including those held off-campus) must show their Murdoch University Student Card to facilitate photographic identification.

**NO OTHER FORM OF IDENTIFICATION WILL BE ACCEPTED.**

**past examination papers**
Past examination papers are available from the library. The web address is: [http://wwwlib.murdoch.edu.au/exams/index.html](http://wwwlib.murdoch.edu.au/exams/index.html). They give you an idea of the standard required and for you to try as practice during your final revision.
However, note that the contents and the types of questions may change from year to year.

**examine your script**
Students may inspect their marked examination scripts and discuss the marking with the Unit Coordinator within 14 days of the posting of results (Degree Regulation 43).

**Final Grade**

In order to pass this unit, you must satisfy the following conditions:

- **achieve an overall aggregate score of 50% or higher for all of the combined assessments (Continuous Assessment and Final Examination);**

  and

- **achieve a satisfactory performance in the final examination. A satisfactory performance is normally considered to be 45% or higher;**

  and

- **achieve a satisfactory performance in the continuous assessment (C Exercises, C Project, Assembly Language Exercises and Assembly Language Program). A satisfactory performance is normally considered to be 45% or higher.**

Failure to achieve the above indicates that the unit objectives have not been met and so the unit has not been successfully completed.

See Section 11.7 and 7.2 of the assessment code regarding grades


**Note that marks may be moderated to ensure equity of marking by different tutors.**

<table>
<thead>
<tr>
<th>Letter</th>
<th>Grade</th>
<th>Percentage Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD</td>
<td>High Distinction</td>
<td>80 - 100</td>
</tr>
<tr>
<td>D</td>
<td>Distinction</td>
<td>70 - 79</td>
</tr>
<tr>
<td>C</td>
<td>Credit</td>
<td>60 - 69</td>
</tr>
<tr>
<td>P</td>
<td>Pass</td>
<td>50 - 59</td>
</tr>
<tr>
<td>N</td>
<td>Fail</td>
<td>Below 50</td>
</tr>
<tr>
<td>DNS</td>
<td>Fail and did not submit any assignments after the HECS census date</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Supplementary Assessment</td>
<td>45 - 49*</td>
</tr>
</tbody>
</table>

*The recommendation for the award of the grade S shall be at the discretion of the Unit Coordinator.*
Appeals

Appeals against the final grade must be submitted in writing to the Executive Dean, Division of Arts, according to the procedures given in Degree Regulation 49.

University Policy on Assessment

Visit the unit welcome page for links to all information relating to assessment policies and grading:


Assessment for this unit is in accordance with the provisions of Degree Regulations 40-48. Check these in the current Murdoch University handbook and Calendar or

http://www.murdoch.edu.au/admin/legslin/regs/bachelor.html#assessment

Assessment roles and responsibilities

Please refer to the University Policy at


Honesty in assessment and avoiding plagiarism

For guidelines on honesty in assessment including avoiding plagiarism, see

http://www.murdoch.edu.au/admin/discipline/

Plagiarism and Collusion

The University regards most seriously any acts of dishonesty relating to assessment. Students are advised to refer to the Unit Welcome Page at

http://www.murdoch.edu.au/vco/secretariat/admin/codes/assessapp2.html#dishonesty

Non-Discriminatory Language

The University is committed to the use of non-discriminatory language in all forms of communication. Students and staff should avoid the use of discriminatory language in units and in all other activities within the University. This applies to both oral and written communication. Further information can be found at:

7. Assignment Submission Requirements

What to submit

Each assignment submission must be in accord with the following general guidelines:

- Assignment should be neatly packaged and contained within a clear A4 plastic envelope or folder. DO NOT STAPLE the opening closed, use one or two paper clips if necessary.
- Internal students must complete a cover sheet which is clearly visible and clearly shows the unit code, your name, student#, tutor name, tutorial time and a signed declaration saying that 'Unless otherwise specified, this assignment is entirely my own work and has not been submitted for assessment in another unit'. An electronic copy of the cover sheet will be available on the unit ftp site. External students have all these details contained on the assignment attachment form they received from the External Studies Office.

- All work to be assessed must be on A4 paper. Any written material must be neat, preferably printed. All material including printouts of programs and test data etc must be of the same (A4) size. Cut them if needed and definitely do not have small pieces of paper less than A4 size.
- A disk or CD (labelled with student name and number, unit code and title, tutor name, and assignment number) with any source code, executable programs, test data input, output etc. The contents of the disk or CD must also contain only the material relevant to that assignment and organised into directories for each assignment question or topic programming problem.

Failure to follow these guidelines will result in marks being deducted.

For each assignment question, the material that should be submitted and is specific to the question, will be described as part of that question. Even if you do not succeed in obtaining a complete working program, you should still submit what you have been able to accomplish, including how you might have tested the program in order to check its correctness.

With regard to the actual program, you should pay particular attention to the following points as these will be taken into account when marking your submissions:

- General layout of the program,
- Your name and purpose comment at the beginning of program,
- Useful comments throughout the program,
- Meaningful identifier names
- Modular structure of program
- Your comments on your program’s performance/limitations
Internal Students

The assignments should be placed in the brown wooden boxes with the ICT106 label, situated in level 3 of the ECL building outside of the School of IT office. Note that there are many boxes corresponding to different units. **IT IS YOUR RESPONSIBILITY NOT OURS** to place your assignment in the correct box.

External Students: Practice Work Exercises and Disks

Submission of the practice work exercises does not require the same amount of material to be submitted nor the detail required for that corresponding to the assignment programming problems. The exercise may only require a small written answer and/or the development of a small program (with any necessary data). Hard copy (printed copy) of any program or written work to be assessed should be submitted as well as a copy on disk or CD.

Disk or CD can be submitted by post in ordinary envelopes, using the size that accommodates A4 pages folded once. Cut two pieces of stiff cardboard slightly larger than the size of the disk, place the diskette within, tape lightly, and submit in the envelope. Disk or CD cover boxes are available from Australia Post. Ensure that your disk is clearly and fully labelled.

External Students - Assignment Attachment Sheet

External students must include, with each submission, an Assignment Attachment having sections A, B and C completed. Mail or deliver your assignments to the External Studies Office, and **not** to your tutor **unless arranged with your tutor**. (Consult your tutor to confirm the preferred means of submission.)

All Students

**IT IS OF THE UTMOST IMPORTANCE** that you retain a complete copy of all work submitted for assessment until completion of the unit.
8. **Teaching Week Table**

The table below lists the dates corresponding to the start of each teaching week (Monday) in the second semester. There are two teaching breaks as shown.

<table>
<thead>
<tr>
<th>Week Number</th>
<th>Semester 2, 2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24 Jul</td>
</tr>
<tr>
<td>2</td>
<td>31 Jul</td>
</tr>
<tr>
<td>3</td>
<td>7 Aug</td>
</tr>
<tr>
<td>4</td>
<td>14 Aug</td>
</tr>
<tr>
<td><strong>Non-teaching Week : 21 Aug</strong></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>28 Aug</td>
</tr>
<tr>
<td></td>
<td>(C Ex 1, 6% and Proj Abstract, 5% due this week)</td>
</tr>
<tr>
<td>6</td>
<td>4 Sept</td>
</tr>
<tr>
<td>7</td>
<td>11 Sept</td>
</tr>
<tr>
<td>8</td>
<td>18 Sept</td>
</tr>
<tr>
<td><strong>Non teaching Week: 25 Sept</strong></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2 Oct</td>
</tr>
<tr>
<td></td>
<td>(C Ex 2, 8% due this week)</td>
</tr>
<tr>
<td>10</td>
<td>9 Oct</td>
</tr>
<tr>
<td>11</td>
<td>16 Oct</td>
</tr>
<tr>
<td></td>
<td>(C Project, 15% due this week)</td>
</tr>
<tr>
<td>12</td>
<td>23 Oct</td>
</tr>
<tr>
<td></td>
<td>(Assembly Ex, 6% due this week)</td>
</tr>
<tr>
<td>13</td>
<td>30 Oct</td>
</tr>
<tr>
<td></td>
<td>(Assembly Program, 10% due this week)</td>
</tr>
</tbody>
</table>
9. **Advice on How to Study this Unit**

Some of you will have already developed effective studying habits. What works well for one person may not for another, and so it is important that you find a method that suits you. However, the system described below seems to work well for most people.

**Read this Outline**

Quickly read through the Introduction and Topics in the Study Guide so that you are aware of the topics to be studied and the unit objectives. Note the assessment requirements.

Check on the textbook and the computing equipment and software required. Arrange to get these essential requirements as quickly as possible. If you have difficulty in getting any of them consult the Unit Coordinator.

The unit is organised around 12 Topics. Each topic has the same basic structure. For each topic, you should:

1. Do the required **Reading**. Find additional resources if necessary.
2. Understand the material in the **Objectives** section
3. Download the practical work sheet from the unit ftp site. Attempt and work through them.
4. When required, demonstrate exercises/programs (External students - submit these)

Following is a more detailed description of each of these parts:

1. **Reading**
   
   Read the chapter of the textbook and any other notes or extra reading material provided for that Topic. It is always best to read the textbook chapter lightly on the first reading, and then go back a second time in more detail taking into account the requirements of the Objectives section. Make notes of any items which you find unclear. That way, when you have completed the topic and are still unclear about certain points, you can use this as a reference when talking to your tutor.

   In some cases, the reading material is duplicated. The same concept is presented from more than one source. This has proven helpful to some students who may not understand one source but understand the other.

2. **Objectives**
   
   The objectives section is intended to guide your understanding of the material covered in the Topic. It details the terms and concepts you should be familiar with after completing the topic, along with any skills and methods you should have developed.

   The Objectives section normally contains the following two parts:

   - **Concepts and terms**
     
     These are the concepts and terms you should be familiar with after completing the Topic. When you have completed the reading for the Topic, you should check your understanding of the material by writing the definitions of all the concepts and terms
listed. You can copy these definitions directly from the textbook into the study guide as an aid to revision.

- **Skills**
  Section 2 lists the things you should be able to do at the end of the Topic. Typically these include items such as understanding various representations, methods of storage or accessing information, assembly language instructions, constructs etc. Learn to experiment and try out small program fragments in order to sort out any problems in your understanding of the material. Where appropriate, always attempt any practical exercises on the computer.

3. **Practical Work**
   Nearly all of the topics have practice exercises prescribed. These will be placed at the unit's ftp site progressively during the semester. Some of the practice exercises require just thinking, while others require small programs to be written. These exercises represent an important component of the unit so spend an appropriate amount of time on this section of the work. Completion of these exercises is a good guide to your understanding of the material. As you work through these you will gain experience with the operating system and compiler and develop your skills in programming and debugging. If you get stuck on a problem, it may pay to lay it aside and come back to it another day. If you receive comments from your tutor, use them to review your work. If in doubt, ask the tutor for further explanations.

   **Internal students** will have to demonstrate the Practice Exercises during tutorials while **External students** must formally submit the Practice Exercises.

4. **Assignments**
   **All students** (both internal and external) must submit solutions to the exercises and the project. The exercises and project sheet will be placed at the unit's ftp site during the semester. If you are unable to access the files from the ftp site, please get in touch with your tutor or unit coordinator as soon as possible.

   Presentation of the assignment work must be in accord with that described in the Submission Requirements section contained in this study guide.

   You will be given a letter grade for the assignments.
Contact Your Tutor
You should not spend an unreasonably long time on one problem. Go on to something else and get in touch with your tutor for some help. Help will be readily available provided you demonstrate to your tutor that you have made a reasonable effort. Whether an internal or external student, it always pays to prepare a list of questions you wish to ask and perhaps some of the work done thus far. When you telephone (or fax or email) your tutor, you should have your Study Guide, Unit Notes and the list of questions with you, so that you do not waste time (and money) searching for information during your conversation with the tutor.

STUDY PACE
The required work is exactly the same for both internal and external students. External students will proceed at their own pace through the unit, using the unit organisation table section and the submission dates as a guide to maintaining an adequate rate of progress through the unit. Internal students have the advantage of lectures and tutorials to pace themselves through the unit. External students should access the lecture notes from the ftp site.

As you study and work through the unit, remember to balance the two complementary aspects: theory versus actually using the computer. Too often, students are more concerned with using the computer than understanding some of the fundamental concepts.

Plan for a time allocation of at least 10 hours per week to your study of this unit. It is best if you maintain a constant, steady rate of progress through the unit. Note that this is not a unit in which all of your study can be left to the last moment. Additionally, attempt each of the practical exercises at the end of the topics immediately after working through that topic. It may be too late to attempt the work immediately prior to when the assignment is due.

All that remains is for me to wish you all the best as you work your way through the unit. I sincerely hope you will find the time you spend studying this unit interesting as well as rewarding.

Complete the following information yourself:

Tutor: ____________________________________________________________

Office: ____________________________

Phone: ____________________________

Email: ____________________________