# Course Outline

# COMP2121
# Microprocessors and Interfacing
### Teaching Period 2 2009

## Contents


- Course Aims
- Student Learning Outcomes
- Assumed Knowledge
- Course Staff
- Course Details
- Resources for Students
- Course Schedule
- Teaching Strategies
- Assessment
- Academic Honesty and Plagiarism
- Course Evaluation and Development


## Course Aims

This course aims to introduce important topics in microprocessors and microcontrollers. If you successfully complete this course, you will have a working knowledge of microprocessors and interfacing, and will be able to solve problems related to the topics covered in this course, and will be proficient in AVR assembly programming.

By the end of this subject you will be able to:

- understand how numbers and characters are represented in computers
- understand basic digital components
- apply general concepts of Instruction Set Architectures (ISAs)
- interpret a number of ISAs including the AVR
- independently write assembler programs
- interface a microcontroller with typical low level I/O systems
- build a small working embedded system

## Student Learning Outcomes

This course contributes to the development of the following graduate attributes:

| **Graduate Attribute** | Where Acquired |
|---|---|
| the skills involved in scholarly enquiry | Coursewide |
| an in-depth engagement with relevant disciplinary knowledge in its interdisciplinary context | Coursewide |
| the capacity for analytical and critical thinking and for creative problem solving | Labs and Project |
| the ability to engage in independent and reflective learning | Coursewide |
| the skills to locate, evaluate and use relevant information (Information Literacy) | Assignment |
| the capacity for enterprise, initiative and creativity | Labs and Project |
| an appreciation of, and a responsiveness to, change | Project |
| a respect for ethical practice and social responsibility | Coursewide |

## Assumed Knowledge

Before commencing this course, students should:

- be able to competently program in C;
- be able to represent integers in binary;
- be able to work independently; and
- be able to work in a group;

These are assumed to have been acquired in the courses COMP1021 or COMP1091 or COMP1721 or COMP1921 or COMP1927 .

## Course Staff

**Contact Details**

| Staff Name | Role | Email | Phone |
|---|---|---|---|
| Sri Parameswaran | Course Convenor | sridevan at cse.unsw.edu.au | 57223 |

**Consultations**

| Staff Name | Day/Time | Location |
|---|---|---|

# Course Details

**Course Code:** COMP2121
**Course Title:** Microprocessors and Interfacing
**Units of Credit:** 6
**Course WebSite:** http://www.cse.unsw.edu.au/~cs2121/
**Handbook Entry:** http://www.handbook.unsw.edu.au/undergraduate/courses/2009/COMP2121.html

# Resources for Students

Text Book

- Fredrick M. Cady: Microcontrollers and Microcomputers Â—Principles of Software and Hardware Engineering.

Further References:

1. Mano, M.M. & Kime, C.R.: Logic and Computer Design Fundamentals, 3rd Edition, Prentice-Hall (2003).
2. Brian Kernighan & Dennis Ritchie: The C Programming Language, 2nd Ed., Prentice Hall, 1988, ISBN: 0-13-110362-8
3. David Patterson and John Hennessy: Computer Organization & Design: The HW/SW Interface," 2nd Ed Morgan Kaufmann, 1998, ISBN: 1 - 55860 - 491X

# Course Schedule

p>Programmer model of computer organization using assembly and machine language. Process of translation from high-level language to machine instructions. Number representation, computer arithmetic, instruction sets, I/O interfacing, I/O interrupts, buses, memory systems, serial communication and analog input and output.

# Teaching Strategies

In this course a variety of Teaching strategies will be used.

1. Teaching in class, where lectures are delivered and interaction encouraged.
2. Laboratory exercises are performed under supervision and assessed.
3. An assignment and a project are given which enable you will expand upon the concepts already learned, or will apply concepts learned to create an artifact.
4. A number of tutorials will be regularly posted, which you can do in your own time.

You are expected to spend an hour on the course for every hour you spend in class (i.e., 5 hours per week). You should practice programming as much as possible.

Creativity in subjects such as this will only be possible when you are an expert. It is no different from being a pianist, artist, or a mathematician.

You should understand all introduced concepts clearly. If not, you are encouraged to stop and ask.

# Assessment

There will be an assignment and a project for this course each counting towards 25% of your final grade.

- Assignment (10%)
- Project (15%)

The assessment for this course will consist of: a 3 hour final exam which will be worth 50%; a mid semester exam will count towards 10% of your final mark; Labs will count towards 15% of your final mark; and the aforementioned assignments will make up the last 25%. The final examination examine whether you:

- understand how numbers and characters are represented in computers
- understand basic digital components
- can apply general concepts of Instruction Set Architectures (ISAs)
- can interpret a number of ISAs including the AVR

- can independently write assembler programs

The first assignment will examine your ability to interpret other ISAs The second assignment will examine your ability to build a small working embedded system The labs will examine whether you can:

- independently write assembler programs
- interface a microcontroller with with typical low level I/O
- build a small working embedded system

## Academic Honesty and Plagiarism

**Plagiarism** is defined as "using the words or ideas of others and presenting them as your own". UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Learning Centre: Plagiarism and Academic Integrity
- MyUNSW: Plagiarism and Academic Misconduct
- CSE: Addendum to UNSW Plagiarism Guidelines
- CSE: Yellow Form (whose terms you have agreed to)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism.

## Course Evaluation and Development

This course is evaluated each session using the CATEI system.

In the previous offering of this courses, students noted ...

Based on their comments, we have ...