MTRN2500
Computing for Mechatronic Engineers
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. STAFF CONTACT DETAILS</td>
<td>3</td>
</tr>
<tr>
<td>2. COURSE DETAILS</td>
<td>3</td>
</tr>
<tr>
<td>3. RATIONALE FOR INCLUSION OF CONTENT AND TEACHING APPROACH</td>
<td>6</td>
</tr>
<tr>
<td>4. TEACHING STRATEGIES</td>
<td>6</td>
</tr>
<tr>
<td>5. ASSESSMENT</td>
<td>7</td>
</tr>
<tr>
<td>6. ACADEMIC HONESTY AND PLAGIARISM</td>
<td>9</td>
</tr>
<tr>
<td>7. COURSE SCHEDULE</td>
<td>10</td>
</tr>
<tr>
<td>8. RESOURCES FOR STUDENTS</td>
<td>10</td>
</tr>
<tr>
<td>9. COURSE EVALUATION AND DEVELOPMENT</td>
<td>11</td>
</tr>
<tr>
<td>10. ADMINISTRATIVE MATTERS</td>
<td>11</td>
</tr>
</tbody>
</table>
1. STAFF CONTACT DETAILS

Contact details and consultation times for course convener

A/Prof. Jay Katupitiya
Room 304 (Tyree Energy Technology Building)
Tel (02) 9385 4096
Fax (02) 9663 1222
Email j.katupitiya@unsw.edu.au

Consultation concerning this course is available on Monday–1600-1700 hours at the above mentioned office.

Contact details and consultation times for additional lecturers and practice session/laboratory teaching staff

Dr. Jose Guivant
Room 303 (Tyree Energy Technology Building)
Tel (02) 9385 9820
Fax (02) 9663 1222
Email j.guivant@unsw.edu.au

2. COURSE DETAILS

Units of credit

This is a 6 units-of-credit (UoC) course and involves 6 hours per week face-to-face contact. There will be one 2 hour lecture and one 1 hour practice session a week conducted by the academic in charge, for each student. In addition, 3 hours of computer laboratory time is set aside for each student to help acquire programming/interfacing skills.

UNSW expects that you will put in on average 40 h/w for 24 UoC (including both in-class and out-of-class time) for an effective 14 weeks of the session (12 weeks plus stuvac plus one effective exam week) for an average student aiming for a credit grade. Various factors, such as your own ability, your target grade, etc., will influence the time needed in your case. Some students spend much more than 40 h/w, but you should aim for not less than 40 h/w on coursework for 24 UoC.

This means that you should aim to spend not less than 10 h/w on this course, i.e. an additional 4 h/w of your own time. This should be spent in making sure that you
understand the lecture material, practice questions, further reading about the course material, and revising and learning for the examinations.

**Parallel Teaching**

There is no parallel teaching in this course.

**Summary of the Course**

This course focuses on the continued learning of C and then a migration onto C++ programming for Mechatronic Systems. During the second half of the course, the students are given a thorough understanding of the Matlab tools.

**Aims of the Course**

Description: The first part of this course will develop the programming skills of the students. Their knowledge of C will be further strengthened and will be further developed to acquire skills of object oriented programming using C++. Their assignment tasks will be centred on programming graphical object on the screen using OpenGL and then manipulating them by some means associated with hardware such as joysticks.

The second part of the course aims to give the students an exposure to Matlab starting from fundamental methods ending up in solving complex computational problems including solutions to sets of ordinary and partial differential equations applicable to Mechatronic Systems.

The courses in the Mechatronics discipline are built up on four different areas. They are; mechanical design, computing, electronics and microprocessors, and control systems. The latter three areas are interrelated and this course forms a corner stone of the fundamental courses on which the Mechatronic Engineering course at UNSW is built upon. A high level of programming skills is necessary to develop customised interface routines to communicate with/control various elements of Mechatronic systems. This knowledge is essential in programming control systems and developing software modules for the interfacing of various hardware elements together to form complete Mechatronic Systems. As such the contributions from this course to the Mechatronic Engineering degree program are absolutely essential and vital.

**Student learning outcomes**

At the conclusion of this course, it is expected that you will be able to:

- be well versed with structured and modular programming using C/C++.
- have appreciated the use of software to communicate with external devices.
- be able to understand how to interface to an external device through a computer program to effect control action.
• be able to develop prototype user interfaces to assist in the development of controlled Mechatronic systems.
• have developed a fundamental knowledge of Matlab tools that are important in designing Mechatronic systems.
• Have developed a skill to choose and use Matlab tools to design Mechatronic Systems.

Graduate attributes

UNSW’s graduate attributes are shown at https://my.unsw.edu.au/student/atoz/GraduateAttributes.html

UNSW aspires to develop graduates who are rigorous scholars, capable of leadership and professional practice in a global community. The university has, thus, articulated the following Graduate Attributes as desired learning outcomes for ALL UNSW students.

UNSW graduates will be

1. Scholars who are:
   (a) understanding of their discipline in its interdisciplinary context ✓
   (b) capable of independent and collaborative enquiry ✓
   (c) rigorous in their analysis, critique, and reflection ✓
   (d) able to apply their knowledge and skills to solving problems ✓
   (e) ethical practitioners ✓
   (f) capable of effective communication ✓
   (g) information literate ✓
   (h) digitally literate ✓

2. Leaders who are:
   (a) enterprising, innovative and creative ✓
   (b) capable of initiating as well as embracing change ✓
   (c) collaborative team workers

3. Professionals who are:
   (a) capable of independent, self-directed practice ✓
   (b) capable of lifelong learning ✓
   (c) capable of operating within an agreed Code of Practice

4. Global Citizens who are:
   (a) capable of applying their discipline in local, national and international contexts ✓
   (b) culturally aware and capable of respecting diversity and acting in socially just/responsible ways ✓
   (c) capable of environmental responsibility ✓

✓ = Developed in this course
In this course, you will be encouraged to develop Graduate Attributes 1(a), 1(b), 1(d), 1(f), 2(a), 3(a), and 4(a) by undertaking the selected activities and knowledge content. These attributes will be assessed within the prescribed assessment tasks, as shown in the assessment table on Page 7.

You will be supported in developing the above attributes through:
(i) the design of academic programs;
(ii) course planning and documentation;
(iii) learning and teaching strategies; and
(iv) assessment strategies.

3. **RATIONALE FOR INCLUSION OF CONTENT AND TEACHING APPROACH**

Our primary goal is to provide the students a learning atmosphere within which knowledge dissemination by the lecturer and knowledge extraction by the student is facilitated. Within this atmosphere the student will be given a sound theoretical basis for the subject matter. They will also be provided stimuli and resources that they can use to extract further knowledge outside the classroom. This will further be enhanced by the compulsory laboratory sessions in which the students will be guided to develop their own (i) software modules to represent objects in an object-oriented programming setup, (ii) ability to develop user interfaces for Mechatronic system. Students learn most effectively when they are confronted with practical systems that relate to lecture content. This will be achieved in this course by requiring compulsory laboratory exercises.

Effective learning requires a system that demands problem solving by students rather than problems solved by the teacher. To facilitate the development of problem solving, the students are provided ample guidance to develop solutions to problems. The essential attributes are the desire to continuously develop the programming skills, especially directed at interfacing.

4. **TEACHING STRATEGIES**

Teaching of this course is through lectures and laboratory sessions. All laboratory work is individual work and attendance is preferred.

The provision of the learning environment in the laboratory is to facilitate you to develop confidence in managing laboratory tasks as projects. Demonstrators in the laboratories are there to provide you all the guidance and assistance is managing the laboratory tasks.
5.  **ASSESSMENT**

**General**

You will be assessed by way of laboratory based and non-laboratory based assignments and an examination, some of which involve demonstration of the execution of your programs.

The breakdown of the marks awarded are given below.

<table>
<thead>
<tr>
<th></th>
<th>Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments</td>
<td>45 marks</td>
</tr>
<tr>
<td>End of session examination</td>
<td>55 marks</td>
</tr>
<tr>
<td>Total Mark</td>
<td>100 marks</td>
</tr>
<tr>
<td>Total mark required to pass the course</td>
<td>50 marks</td>
</tr>
</tbody>
</table>

**Laboratory Exercises**

The set assignments to be assessed are listed in the table below.

<table>
<thead>
<tr>
<th>No.</th>
<th>Assignment</th>
<th>Mark</th>
<th>Learning outcomes assessed</th>
<th>Graduate attrib assessed</th>
<th>Due Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Programming a Joystick</td>
<td>10</td>
<td>C Programming and hardware interfacing</td>
<td>1(a) 1(b) 1(d) 1(f)</td>
<td>Week 4</td>
</tr>
<tr>
<td>2</td>
<td>Rendering Graphical objects</td>
<td>15</td>
<td>C++ and user interface design</td>
<td>1(a) 1(b) 1(d) 1(f) 2(a) 3(a)</td>
<td>Week 7</td>
</tr>
<tr>
<td>3</td>
<td>Matlab assignment</td>
<td>20</td>
<td>Real time systems</td>
<td>1(a) 1(b) 1(d) 1(f) 2(a) 3(a)</td>
<td>Week 11</td>
</tr>
</tbody>
</table>
Presentation

All submissions should have a standard School cover sheet available on the School website at [www.engineering.unsw.edu.au/mechanical-engineering/forms-and-guidelines](http://www.engineering.unsw.edu.au/mechanical-engineering/forms-and-guidelines). All submissions are expected to be neat, and clearly set out. All calculations should be shown as, in the event of incorrect answers, marks are awarded for method and understanding.

The preferred set-out of any numerical calculation is similar to the following:

\[ \Delta = \rho \nabla \]  
(Equation in symbols)

\[ = 1.025 \times 200 \]  
(Numbers substituted)

\[ = 205 \text{ t} \]  
(Answer with units)

Submission

All assignments will include complete and operational software development. All students are required to sit with a demonstrator one on one to explain their programs and to demonstrate the execution of their programs. The explanation and the verbal questions by the demonstrator on your program will form the basis for your assessment.

All laboratory reports must be submitted online strictly according to the instructions given in the assignment specification. Late submission of laboratory reports/assignments attracts a penalty as described in the respective laboratory instruction sheets/assignment specifications.

For more information on submission of assignments, see [Administrative Matters for All Courses](http://www.engineering.unsw.edu.au/mechanical-engineering/forms-and-guidelines) available on the School website.

Criteria

A detailed marking criteria is included in the assignment specification sheet. The following areas must be highlighted in your reports/submissions.

- Logical layout of a pseudo code.
- Structured programming.
- Modular software composition.
- Efficient software constructs.
- Operational software.

Examination

There will be one two-hour examination at the end of the session, covering all material taught in the whole session.

You may need to provide your own calculator, of a make and model approved by UNSW, for the examination. The list of approved calculators is shown at [https://student.unsw.edu.au/exam-approved-calculators-and-computers](https://student.unsw.edu.au/exam-approved-calculators-and-computers)

It is your responsibility to ensure that your calculator is of an approved make and model, and to obtain an “Approved” sticker for it from the School Office or the
Engineering Student Centre prior to the examination. Calculators not bearing an “Approved” sticker will not be allowed into the examination room.

**Special Consideration and Supplementary Assessment**

For details of applying for special consideration and conditions for the award of supplementary assessment, see *Administrative Matters for All Courses*, available from the School website.

**6. ACADEMIC HONESTY AND PLAGIARISM**

Plagiarism is using the words or ideas of others and presenting them as your own. Plagiarism is a type of intellectual theft. It can take many forms, from deliberate cheating to accidentally copying from a source without acknowledgement. UNSW has produced a booklet which provides essential information for avoiding plagiarism: [https://my.unsw.edu.au/student/academiclife/Plagiarism.pdf](https://my.unsw.edu.au/student/academiclife/Plagiarism.pdf)

There is a range of resources to support students to avoid plagiarism. The Learning Centre assists students with understanding academic integrity and how not to plagiarise. They also hold workshops and can help students one-on-one. Information is available on the dedicated website Plagiarism and Academic Integrity website: [http://www.lc.unsw.edu.au/plagiarism/index.html](http://www.lc.unsw.edu.au/plagiarism/index.html)

You are also reminded that careful time management is an important part of study and one of the identified causes of plagiarism is poor time management. Students should allow sufficient time for research, drafting and the proper referencing of sources in preparing all assessment tasks.

If plagiarism is found in your work when you are in first year, your lecturer will offer you assistance to improve your academic skills. They may ask you to look at some online resources, attend the Learning Centre, or sometimes resubmit your work with the problem fixed. However more serious instances in first year, such as stealing another student’s work or paying someone to do your work, may be investigated under the Student Misconduct Procedures.

Repeated plagiarism (even in first year), plagiarism after first year, or serious instances, may also be investigated under the Student Misconduct Procedures. The penalties under the procedures can include a reduction in marks, failing a course or for the most serious matters (like plagiarism in a honours thesis) even suspension from the university. The Student Misconduct Procedures are available here: [http://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf](http://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)

Further information on School policy and procedures in the event of plagiarism is presented in a School handout, *Administrative Matters for All Courses*, available from the School website.
7. COURSE SCHEDULE

Lectures: Monday 1600-1800 hours, in Rex Vowels Theatre.
Tuesday 1400-1500 hours, in Webster A.
Laboratory classes: Block house
Prerequisites: COMP1911/COMP1917

<table>
<thead>
<tr>
<th>Week</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Introduction</strong>: Programming needs of Mechatronic Systems. Introduction to the course. Software platforms used for programming within the course. Introduction to Object Oriented Programming.</td>
</tr>
<tr>
<td>2</td>
<td><strong>C Programming Revision</strong>: Editing, Compiling, Linking, Debugging, Loading and executing. Structures and classes. Creation of classes and their use in programmes.</td>
</tr>
<tr>
<td>3-5</td>
<td><strong>C++ Programming</strong>: Object Oriented Programming. Constructors, destructors, member functions, function overloading, class derivation, abstract classes.</td>
</tr>
<tr>
<td>6</td>
<td><strong>C++ Programming</strong>: Advanced concepts. Pass through objects, streams, reference objects.</td>
</tr>
<tr>
<td>7</td>
<td><strong>Matlab</strong>: Understanding the Matlab Programming Language.</td>
</tr>
<tr>
<td>8</td>
<td><strong>Matlab</strong>: Exploiting Object Programming in Matlab</td>
</tr>
<tr>
<td>9</td>
<td><strong>Matlab</strong>: Using Matlab for processing massive data: images, 3D images, sound, sampled signals.</td>
</tr>
<tr>
<td>10</td>
<td><strong>Matlab</strong>: Hybrid solutions via Mex (C,C++) extensions in Matlab, for efficient and flexible solutions.</td>
</tr>
<tr>
<td>11</td>
<td><strong>Matlab</strong>: Solving a complex real problem, for on-line Processing and Control, via Matlab (a problem related to Mechatronics and Robotics)</td>
</tr>
<tr>
<td>12</td>
<td>Revision</td>
</tr>
</tbody>
</table>

The schedule shown may be subject to change at short notice to suit exigencies.

8. RESOURCES FOR STUDENTS

As part of the course some handouts will be made available to the students during the course by placing them on Blackboard Learn.

Recommended Textbooks

- J.Katupitiya & K. Bentley, "Interfacing with C++", Springer 2006
- P.H. Winston, "On to C", Addison Wesley
- P.H. Winston, "On to C++", Addison Wesley

Additional Readings

A number of additional documents will be available on the web. Some materials from earlier years may also be available at Blackboard's MTRN2500 Home page.
9. COURSE EVALUATION AND DEVELOPMENT

Feedback on the course is gathered periodically using various means, including the Course and Teaching Evaluation and Improvement (CATEI) process, informal discussion in the final practice session for the course, and the School’s Student/Staff meetings. Your feedback is taken seriously, and continual improvements are made to the course based, in part, on such feedback.

10. ADMINISTRATIVE MATTERS

You are expected to have read and be familiar with Administrative Matters for All Courses, available on the School website. This document contains important information on student responsibilities and support, including special consideration, assessment, health and safety, and student equity and diversity.

Jay Katupitiya
11 July 2014