Cranfield UNIVERSITY

SCHOOL OF ENGINEERING

COURSE INFORMATION MANUAL

MSc
Computational & Software Techniques in Engineering

2011/2012

Course Director: Dr Karl Jenkins
Course Administrator: Pauline Buck

www.cranfield.ac.uk/soe/intranet/student

This document should be read in conjunction with the School of Engineering's 'Students' Information Booklet 2011/2012

SCHOOL OF ENGINEERING
MSc COURSE INFORMATION MANUAL

September 2011

Dear Course Member

Welcome to the Applied Mathematics and Computing Group within the School of Engineering (SoE). This document contains information about your course and those available to help you. Please look at it carefully and keep it for future reference. If you have problems please contact the appropriate member of staff or, if more appropriate, the Course Administrator, Pauline Buck, will help.

We shall see quite a lot of each other in the next year and we all look forward not only with working with each other but also to some enjoyable social occasions.

Good luck with your studies and your career development.

Dr Karl Jenkins
(MSc Course Director)

Pauline Buck
(MSc Course Administrator)

ACADEMIC/ SUPPORT STAFF

Head of School of Engineering
Professor Minoo Patel
Head of Applied Maths & Computing Group
Professor Chris Thompson
Building 52/Room 502
Tel ext: 5473
Email: chris.thompson@cranfield.ac.uk
Course Director
Dr Karl Jenkins
Building 52/Room 503
Email: k.w.jenkins@cranfield.ac.uk
Course Administrator
Pauline Buck
Building 52/Room 15
Tel ext: 4746
Email: amac-mscadmin@cranfield.ac.uk

TEACHING/EXTERNAL STAFF:
Professor Chris Thompson (Head of Department)
Building 52/Room 18
Email: chris.thompson@cranfield.ac.uk

Professor Leonid Gelman
Building 52/Room 504
Email: l.gelman@cranfield.ac.uk

Dr Toby Breckon
Building 52/Room 501
Email: t.breckon@cranfield.ac.uk

Dr Carol Armitage
Building 52/Room 506
Email: c.armitage@cranfield.ac.uk

Dr Stuart Barnes
Building 52/Room 309
Email: s.barnes@cranfield.ac.uk

Mr Tony Lawrence
Cranfield Computer Centre
Tel ext: 4206
Email: a.lawrence@cranfield.ac.uk

Dr Peter Sherar
External
Email: p.sherar@cranfield.ac.uk

Mrs Kath Tipping
External
Email: k.tipping@cranfield.ac.uk

1. COURSE STRUCTURE

Core Modules

C Programming (pre-requisite)

- Computational Methods
- C++ Programming
- Computer Graphics (*half module*)
- Management for Technology

Module Options in CAE

- Geometric Modelling (*one and a half modules*)
- CAE Applications & PLM
- Advanced Engineering Analysis
- Computational Engineering (fluids)
- Advanced CAE Applications
- Computational Engineering Design Optimisation (*half module*)
- Advanced Graphics (*half module*)

MSc Mark Range & Standard	Criteria / Descriptor
80% - 100% **Standard = Excellent**	Demonstrating a comprehensive knowledge and understanding of the subject and subfields. High capacity for critical evaluation. Novel application of the subject matter to a specific context. **Process:-** Requiring a student to have: Undertaken extensive further reading. Produced a well structured piece of work. Demonstrated excellent communication skills. Exercised a high level of original thought.
70% - 79% **Standard = Very Good**	Demonstrating an extensive knowledge and understanding of the subject and subfields. Very good capacity for critical evaluation. Effective application of the subject matter to a specific context. **Process:-** Requiring a student to have: Undertaken substantial further reading. Produced a well structured piece of work. Demonstrated very good communication skills. Exercised a significant level of original thought.
60% - 69% **Standard = Good**	Demonstrating a good knowledge and understanding of the subject and subfields. Good capacity for critical evaluation. Competent application of the subject matter to a specific context. **Process :-** Requiring a student to have: Undertaken some further reading. Produced a well structured piece of work. Demonstrated good communication skills.
50% - 59% **Standard = Satisfactory**	Demonstrating a satisfactory knowledge and understanding of the subject and subfields. Standard critique of the subject matter. Adequate application of the subject matter to a specific context. **Process:-** Requiring a student to have: Undertaken adequate further reading. Produced an adequately structured piece of work. Demonstrated basic but satisfactory communication skills.
40% - 49% **Standard = Poor**	Demonstrating an inadequate knowledge and understanding of the subject and subfields. Lacking critique of the subject matter. Limited application of the subject matter to a specific context. **Process:-** Requiring a student to have: Undertaken some relevant reading. Produced a piece of work with a simple structure. Demonstrated marginal communication skills
0% - 39% **Standard = Very Poor**	Demonstrating a lack of knowledge and understanding of the subject and subfields. Absence of critique of the subject matter. Lacking application of the subject matter to a specific context. **Process:-** Requiring a student to have: Undertaken inadequate reading. Produced a poorly structured piece of work. Demonstrated poor communication skills.

2. BRIEF SYLLABI OF LECTURE PROGRAMMES

Module title **C Programming (pre-requisite)**			
Name of module convenor/leader/coordinator **Dr Peter Sherar**			
(a) class contact hours 30	(b) private study hours 70	(c) Total notional hours (i.e. the sum of (a) and (b) 100	Credit rating 0
Assessment method **(not assessed)**			
Prerequisites None			

<u>Aim</u>

The module aims to cover all the main elements making up the C programming language and to provide many illustrative examples of their use in practice. Use is made of 'hands-on' workshops which enable the student to gain confidence with the language and form a preparation for the practical assignment which forms a major part of the course. An Ansi-standard C compiler and development environment is employed.

<u>Syllabus/curriculum</u>

Variables, operators and expressions, Statements and flow control, Functions Pointers and arrays, Strings, Structures and other derived types, Dynamic memory Allocation, Input and output

<u>Intended learning outcomes</u>
On completion of this module the student will be able to:

- Demonstrate a good understanding of the main elements making up a procedural language.
- Use functions, pointers and structures in the C language.
- Write a C program of moderate complexity given a formal specification.
- Understand the concept of dynamic memory allocation and be able to use it in practical applications.
- Apply knowledge and skills to the implementation of advanced data structures in the C language.

<table>
<tr><td colspan="3">Module title Computer Graphics</td><td colspan="2"></td></tr>
<tr><td colspan="3">Name of module convenor/leader/coordinator
Dr Peter Sherar</td><td colspan="2"></td></tr>
<tr><td>(a) class contact hours 15</td><td>(b) private study hours 35</td><td colspan="2">(c) Total notional hours (i.e. the sum of (a) and (b) 50</td><td>Credit rating 5</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td colspan="3">Assessment method Assignment (100%)</td><td colspan="2">Compulsory</td></tr>
<tr><td colspan="3">Prerequisites C++ Programming</td><td colspan="2"></td></tr>
</table>

Aim

The aim of this half module is to provide the student with a hands-on introduction to the programming paradigms, techniques and libraries used in the construction of graphical user interfaces. It covers the model, view, controller (MVC) paradigm and accompanying GUI programming models used in a number of popular user interface libraries. On the practical side it aims to provide the student with skills in GUI construction using Windows Forms under the .NET framework in C++.

The module also provides an overview of the mathematical principles behind 2D and 3D visualisation and the viewing pipeline and their practical implementation in the widely used Open-GL graphics library. Some representative GUI based 2D and 3D Open-GL applications using Windows Forms are developed.

Syllabus/curriculum

Programming models for GUI development – MVC, event handling and GUI component libraries , Windows Forms and .NET, Mathematical principles behind 2D and 3D visualisation – the viewing pipeline, The Open-GL graphics library, Development of CG applications using Open-GL and Windows Forms

Intended learning outcomes

On completion of this module the student will be able to:

- Understand the principal programming paradigms and models underpinning modern user interface libraries.
- Apply these principles in the development of basic GUI applications using the Windows Forms windowing toolkit.
- Understand the mathematical principles behind 2D and 3D visualisation and their implementation in Open-GL.
- Develop basic graphical based applications using Open-GL, either in standalone mode or with Windows Forms.

Module title **Computational Methods**			
Name of module convenor/leader/coordinator **Kath Tipping**			
(a) class contact hours 30	(b) private study hours 70	(c) Total notional hours (i.e. the sum of (a) and (b) 100	Credit rating 10
Assessment method **Exam (100%)**		Compulsory	
Prerequisites None			

Aim

The module aims to emphasise the importance of linear systems in engineering situations and to provide a variety of computational methods for solving linear systems of equations and eigenvalue problems.

Syllabus/curriculum

Matrices and types of linear systems, Direct elimination methods, Solution to tridiagonal Systems, Conditioning and stability of solutions, Iterative methods and convergence Criteria, Eigenvalue and eigenvector problems

Intended learning outcomes

On completion of this module the student will be able to:

- Understand and apply matrix algebra, linear dependence and independence.
- Choose an appropriate method for solving a particular linear system of equations.
- Demonstrate an awareness of difficulties such as ill-conditioning, and be able to suggest ways of minimising the problems.
- Understand the concept of eigenvalues and eigenvectors and their importance.
- Apply several techniques for finding eigenvalues and eigenvectors in practical problems.
- Use MATLAB to solve some practical problems.

Module title **C++ Programming**			
Name of module convenor/leader/coordinator **Dr Peter Sherar**			
(a) class contact hours 30	(b) private study hours 70	(c) Total notional hours (i.e. the sum of (a) and (b) 100	Credit rating 10
Assessment method **Exam (50%), Assignment (50%)**		Compulsory	
Prerequisites **C Programming**			

Aim

Object oriented programming (OOP) is the standard programming methodology used in nearly all fields of major software construction today, including CAD/CAM and DSIP. In practice, C++ is the most heavily used OOP language. This module aims to answer the question 'what is object oriented programming?', and then looks in detail at the C++ language. Hands-on programming and an assignment form an important part of the course.

Syllabus/curriculum

The OOP methodology and method, Abstraction and encapsulation, Classes, Constructors and destructors, Function and operator overloading, Inheritance, polymorphism and virtual functions, Stream input and output, Templates and template based class libraries, Exception handling

Intended learning outcomes

On completion of this module the student will be able to:

- Understand the object oriented programming methodology and the concepts of abstraction and encapsulation.
- Understand and apply the main elements of C++ classes including constructors and destructors, member functions and overloaded operators.
- Understand and apply the principles of combining classes class using inheritance and/or object composition.
- Build C++ programs of moderate complexity given a specification with exception handling.
- Use template based class libraries, particularly for I/O and data structures.

<table>
<tr><td colspan="2">Module title Management For Technology</td><td colspan="2"></td></tr>
<tr><td colspan="2">Name of module convenor/leader/coordinator
School of Management</td><td colspan="2"></td></tr>
<tr><td>(a) class contact hours
30</td><td>(b) private study hours
70</td><td>(c) Total notional hours
(i.e. the sum of (a) and (b)
100</td><td>Credit rating
10</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="2">Assessment method Exam (80%), Assignment (20%)</td><td colspan="2">Compulsory</td></tr>
<tr><td colspan="2">Prerequisites</td><td colspan="2"></td></tr>
</table>

Aim

Apart from technical knowledge, an engineer must also acquire management training. The aim of this module is to provide the student with the essential training to manage a project and its financial aspects. This is an intensive two-week short course run by the Cranfield School of Management and aims to equip students with the management learning tools required in a modern industrial environment.

Syllabus/curriculum

Corporate Planning, Cash flow forecasting, Legal responsibilities, Pricing and profit planning, Project management, Finance and accounting, Industrial relations, Industrial marketing, Management for R & D.

Intended learning outcomes

On completion of this module the student will be able to:

- Demonstrate an awareness of a range management issues
- Understand the needs and requirements of project management
- Demonstrate and awareness and understanding of the financial aspects associated with project planning.

<table>
<tr><td colspan="2">Module title Geometric Modelling</td><td colspan="2"></td></tr>
<tr><td colspan="2">Name of module convenor/leader/coordinator
Dr Peter Sherar</td><td colspan="2"></td></tr>
<tr><td>(a) class contact hours
45</td><td>(b) private study hours
105</td><td>(c) Total notional hours (i.e. the sum of (a) and (b)
150</td><td>Credit rating
15</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="2">Assessment method Exam (50%), Assignment (50%)</td><td colspan="2">Compulsory</td></tr>
<tr><td colspan="2">Prerequisites C Programming</td><td colspan="2"></td></tr>
<tr><td colspan="4"><u>Aim</u>

The aim of this module is to provide the student with the fundamental algorithms, techniques and software used for the construction of parametric curves, surfaces and solids. The techniques covered here form the basis of free-form modelling as used in CAD/CAM systems and more generally in visualisation and computer graphics. The module also aims to develop an understanding of the basic principles of storing and communicating models of product shape and associated information. Hands-on programming exercises and a modelling assignment form part of the course.</td></tr>
<tr><td colspan="4"><u>Syllabus/curriculum</u>

Polynomial and spline interpolation, B-spline curve fitting and construction techniques, B-spline surface fitting and construction techniques, Solid model representation schemes, Boundary representation models, 3D modelling API’s</td></tr>
<tr><td colspan="4"><u>Intended learning outcomes</u>
On completion of this module the student will be able to:
Understand the definition and benefits of use of spline and B-spline functions.
Demonstrate knowledge of the main methods of free-form curve and surface construction used in CAD/CAM.
Develop computer programs to implement free-form curve and surface construction techniques.
Demonstrate a working knowledge of the methods used for computer representations of engineering product.
Understand the mathematical and computer science techniques deployed in the creation of practical 3D geometric modelling software.
Demonstrate a critical awareness of the issues governing the use of 3D geometric modelling in engineering applications software.</td></tr>
</table>

<table>
<tr><td colspan="2">Module title CAE Applications & PLM</td><td colspan="2"></td></tr>
<tr><td colspan="2">Name of module convenor/leader/coordinator
Tony Lawrence</td><td colspan="2"></td></tr>
<tr><td>(a) class contact hours
30</td><td>(b) private study hours
70</td><td>(c) Total notional hours (i.e. the sum of (a) and (b)
100</td><td>Credit rating
10</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="2">Assessment method Assignment (100%)</td><td colspan="2">Compulsory</td></tr>
<tr><td colspan="2">Prerequisites None</td><td colspan="2"></td></tr>
</table>

Aim

The aim of the CAE Solid Modelling module is to introduce students to key concepts, techniques and applications of a 3D Solid Modelling system. Use is made of structured computer based workshops which employ an industry standard system for 3D Solid Modelling. Introductory lectures are reinforced by the ‘hands-on’ approach since each student will be set an exercise to build a CAE Solid Model of a specified component on the course. The module also provides an overview of PLM.

Syllabus/curriculum

Introduction to I-DEAS CAE Solid Modelling Software, Some benefits of using solid modelling and the CAE approach, Different construction methods for 3D geometrical models, Parametric and variational design, Production of drafting setup details from 3D geometrical parts, Modifying parts and features, Product Lifecycle Management

Intended learning outcomes

On completion of this module the student will be able to:

- Understand how a modern CAE Solid Modelling tool is used.
- Generate solid geometrical parts using a variety of basic construction techniques.
- Apply skills necessary to carry out a variety of basic Solid Modelling tasks.
- Generate 2D drawings from 3D geometrical parts.
- Appreciate the role and scope of PLM in product development

Module title **CAE Advanced Applications**			
Name of module convenor/leader/coordinator **Tony Lawrence**			
(a) class contact hours 30	(b) private study hours 70	(c) Total notional hours (i.e. the sum of (a) and (b) 100	Credit rating 10
Assessment method **Assignment (100%)**		Compulsory	
Prerequisites **CAE Applications**			

Aim

This course covers more advanced aspects of CAE, the aim being to introduce students to key concepts and techniques in the use of CAE application software tools. Use is made of structured computer based workshops which employ industry standard systems for CAD through to Engineering Analysis.

Syllabus/curriculum

Introduction to I-DEAS CAE Finite Element Analysis (FEA) Simulation software, CAE FEA Pre- and Post-Processing, Free mesh and Mapped mesh techniques, Quality checks on nodes and elements, Finite element and geometry based boundary conditions, Utilising solids based modelling geometry for downstream CAE FEA, CAE linear statics analysis using the I-DEAS CAE FEA Simulation software, Case Studies.

Intended learning outcomes

On completion of this module the student will be able to:

- Demonstrate an understanding of how modern CAE Analysis tools are used.
- Use free mesh and mapped mesh generation techniques.
- Generate finite element analysis models by using either geometry from the I-DEAS solid modeller or an external CAD system.
- Use the I-DEAS Simulation Analysis module to run linear static analysis modules.

Module title **Advanced Engineering Analysis**			
Name of module convenor/leader/coordinator **Prof Chris Thompson**			
(a) class contact hours 30	(b) private study hours 70	(c) Total notional hours (i.e. the sum of (a) and (b) 100	Credit rating 10
Assessment method **Exam (100%)**		Compulsory	
Prerequisites **Computational Methods**			

Aim

The numerical solutions of partial differential equations are used for simulating physical systems and phenomena and for the investigation of a wide range engineering applications. These numerical solutions may used in engineering design optimisation to explore the implications of design changes. The aim of this course is to provide the student with the mathematical background to the discretisation of partial differential equations using finite element and finite difference approaches, and an insight into methods for their solution along with the vital numerical techniques for the analysis of the solution and numerical errors.

Syllabus/curriculum

Introduction to Simulation, Finite Element Methods, Finite Difference Methods, Numerical Solution to Partial Differential Equations: Parabolic, Elliptic, Hyperbolic Stability Analysis and Truncation Errors, Case Studies

Intended learning outcomes

On completion of this module the student will be able to:

- Understand the mathematical principles of the discretisation methods.
- Identify problems which are suitable for finite element or finite difference solution.
- Demonstrate a working knowledge of numerical solution methods.
- Demonstrate an understanding of stability analysis and numerical errors.

Module title **Computational Engineering (fluids)**			
Name of module convenor/leader/coordinator **Dr Karl Jenkins**			
(a) class contact hours 30	(b) private study hours 70	(c) Total notional hours (i.e. the sum of (a) and (b) 100	Credit rating 10
Assessment method **Assignment (100%)**		Compulsory	
Prerequisites **None**			

Aim

To introduce the techniques and tools for modelling, simulating and analysing realistic computational engineering problems for industrial applications with practical hands on experience of commercial software packages used in industry.

Syllabus/curriculum

Introduction to Computational Engineering, Fundamental equations , The Computational Engineering Process, Fluid Simulation for Computer Graphics, Modelling techniques, Practical sessions

Intended learning outcomes

On completion of this module the student will be able to:

- An understanding of the Computational Engineering Process.
- Understand the governing equations for fluid systems and how to solve them computationally.
- Be able to write code to solve problems and undertake practical problems using commercial software
- Appreciate the wide range of applications using computational engineering for fluids.

<table>
<tr><td colspan="2">Module title Computational Engineering Design Optimisation</td><td colspan="3"></td></tr>
<tr><td colspan="2">Name of module convenor/leader/coordinator
Dr Carol Armitage</td><td colspan="3"></td></tr>
<tr><td>(a) class contact hours
15</td><td>(b) private study hours
35</td><td colspan="2">(c) Total notional hours (i.e. the sum of (a) and (b)
50</td><td>Credit rating
5</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td colspan="2">Assessment method Assignment (100%)</td><td colspan="3">Compulsory</td></tr>
<tr><td colspan="2">Prerequisites None</td><td colspan="3"></td></tr>
<tr><td colspan="5"><u>Aim</u>

The aim of the CEDO module is to introduce the techniques for modelling, simulating and analysing realistic computational engineering problems in the context of design optimisation. The engineering design optimisation process is introduced along with numerical methods used in optimisation.</td></tr>
<tr><td colspan="5"><u>Syllabus/curriculum</u>

Engineering design optimisation process, Optimisation theory, deterministic: optimality criteria, search methods, stochastic: simulated annealing, genetic algorithms, tabu search, Application of design optimisation theory</td></tr>
<tr><td colspan="5"><u>Intended learning outcomes</u>
On completion of this module the student will be able to:
Demonstrate an understanding of the computational engineering process and its applications.
Demonstrate an understanding of the design optimisation process.
Appreciate design optimisation for industrial applications.</td></tr>
</table>

<table>
<tr><td colspan="2">Module title Advanced Graphics</td><td colspan="2"></td></tr>
<tr><td colspan="2">Name of module convenor/leader/coordinator
Dr Stuart Barnes</td><td colspan="2"></td></tr>
<tr><td>(a) class contact hours
15</td><td>(b) private study hours
35</td><td>(c) Total notional hours
(i.e. the sum of (a) and (b)
50</td><td>Credit rating
5</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="2">Assessment method Assignment (100%)</td><td colspan="2">Compulsory</td></tr>
<tr><td colspan="2">Prerequisites Computer Graphics</td><td colspan="2"></td></tr>
</table>

Aim

High performance computer graphics are used in many areas of software application development, and are fundamental to games, entertainment, CAD and scientific visualisation. The aim of this module is to introduce students to the advanced techniques used in the generation of computer graphics. Building on the basic methods of the Introductory course, students will learn how to generate more realistic effects, such as the use of lighting and surface details to create realistic representations of computer generated graphical objects and display them to the screen.

Syllabus/curriculum

Surfaces and Visibility, Geometric and Raster Algorithms, Light, Illumination and Shading, Computer Animation.

Intended learning outcomes
On completion of this module the student will be able to:

- Understand the concepts, underlying principles and operation of a range of advanced computer graphics algorithms and techniques
- Optimize the graphics pipeline by implementing visible surface algorithms, such as hidden surface removal and z-buffering, leading to real-time performance
- Understand the models of interaction between light and materials, as well as being able to demonstrate a practical capability of implementing such methods
- Implement algorithms using the OpenGL graphics library and apply these techniques to solving a specific problem in computer graphics