De Montfort University

Course Template

1. Basic information

- Course Name: Software Engineering
- Course Code: CC103A
- Level (UG, PG): Undergraduate
- Academic Period: 2014
- Faculty: Faculty of Technology
- Department: Computer Engineering and Cyber security
- PMB COMP
- Offered at: DM - DMU Leicester
- Type (single, joint.): SI
- Highest Award : Bachelor of Science (Honours)
- All possible exit awards : Bachelor of Science; Certificate of Higher Education; Diploma of Higher Education; Institutional Undergraduate Credit
- Award notes : The BSc(Hons) Software engineering in sandwich mode has full exemption and CEng/CSci accreditation (partial)
  Condition
  1. Students must pass, at the first attempt, a practical problem-solving project. This applies to students on the five intakes from 2005 to 2009 inclusive and backdated to the 2003 intake.

Professional Body Recognition

- Accreditation by Professional/Statutory body:
  Yes
- Exemption by Professional/Statutory body:
  Yes
- Details
  The British Computer Society

- Modes of attendance: Main MOA: Full-Time
  Other MOA: Part-Time; Year Out/On Placement
- Mode Notes:
- Course leader: David Smallwood

2. Entry Requirements and Profile

Award
BSc /BSc (Hons) Software Engineering

Standard Entry Requirements
Candidates must normally offer one of the following :-
5 passes at GCE/GCSE level including Mathematics and English with
260 points from at least 2 subjects at A2 or AVCE (excluding General Studies), or equivalent AS levels.
Overall merit in an appropriate BTEC National diploma
Merit in an Advanced GNVQ in Information Technology, plus 3 additional units.
Any qualification deemed equivalent to the above
The particular qualifications of each candidate are considered on an individual basis.
A background in science, mathematics or computing is an advantage. Candidates usually show a keen interest in programming.
Candidates who do not possess the normal entry requirements may be considered for exceptional admission.
Applications are welcomed from mature students whose formal qualifications do not match the levels given above, but who have demonstrated the necessary ability to pursue the course and benefit from it.

3. Course Description

Characteristics and Aims

The worlds of technology, business, entertainment and communications are dependent upon computers and computer software. The role of the software engineer is key in the development and support of the systems upon which these industries rely. Software Engineering provides a broad range of technical computing knowledge and skills, but specifically addresses the practical problems of designing and constructing reliable, robust and maintainable software. It does this by the application of rigorous development techniques and engineering principles based on fundamentals such as logic and theories of programming.

The Quality Assurance Agency (QAA) Computing Benchmarks (2007) identified thirty-seven cognate areas within computing ranging from technical infrastructure through to the professional, social and ethical issues. Software Engineering has been designed as a specialist pathway through the Computer Science course. Topics covered include contemporary computing theory, current methodologies, and a range of practical skills.

Level 4 (First Year)

The first year of study comprises a suite of modules covering the key subject content including the underpinning foundations of computing; computer architecture; programming; web interface development; mathematics for computing; database fundamentals; computational thinking. These modules provide a sound basis for students to make choices for further study at level 2.

Level 5 (Second Year)

The second year of study has a an emphasis on design and development . In particular students are introduced to object-oriented software design and development; database design and implementation; internet software development; and data structures and algorithms.

Industrial Placement

The course offers students the opportunity to apply for a year-long industrial placement. This is often the first time the student sees how the concepts taught on the course are actually used in the "real world". Of course, the particular experience will depend upon the host company but students will often benefit from specialist training and will develop their range of practical, social and interpersonal skills. Students will be able to see first hand what working in a real company is like and will gain personal understanding of the expectations and responsibilities of being an employee in a professional environment.
.

Level 6 (Final Year)

The final year project develops the project management, report writing and presentation skills particularly cherished by many employers. As well as the compulsory project the final year requires students to undertake modules covering software testing, software development methods, software management, software quality assurance, ethics and professional practice. Furthermore, as Software Engineering specialists students undertake a module on Formal Systems in which rigorous approaches to software engineering are introduced. The remainder of the final year (25%) is left for the student to choose modules from a range of options.

Resources

Throughout the course students have access to high quality, modern computer hardware and software. An enthusiastic team of tutors supports learning.

Employment

The increasing world wide demand for computer software that is effective, reliable and maintainable requires specialists. Typically graduates seek work in the design and construction of software in a range of fields including multimedia, networks and communications, internet applications, the games industry, virtual reality and simulation, real-time systems, distributed systems, large database systems, financial and commercial systems.

Teaching, Learning and Assessment Strategies

The Learning Strategies section of the computing subject template describes the range of teaching, learning and assessment styles and strategies within the subject area.
The learning strategies of the course are an amalgamation of the learning strategies of the individual modules on the course.
The compulsory modules on this course ensure that students learn by a combination of practical experience, self-study and research.

The strategies by which students may learn on this course include
· staff directed learning via lectures and tutorials or laboratories.
· Student-centred resource based learning (including web-based resources)
· collaborative and group based
· individual learning
· student centred learning via research
Assessment in each module is designed to meet the specified learning outcomes of the module.
Methods of assessment for modules will include
· time constrained phase tests
· portfolios of work
· programming and other laboratory exercises
· viva voce examinations
· individual and group work
· formal examinations
· project work
reports and presentations

4. Outcomes

Generic outcome headings	What a student should know and be able to do upon completion of the course
• **Knowledge & understanding**	Students should be able to demonstrate a systematic understanding in the following key fields: Computer Architecture Computer Programming Formal Methods and Description Techniques Data Structures and Algorithms Data Analysis Database Design and Implementation Operating Systems and Computer Networks Web Interface Development Human-Computer Interaction Systems Analysis and Design Testing and Evaluation Professionalism and Ethics Students should be able to apply, appropriately, the theoretical knowledge and practical experience they have learnt to the analysis, design, implementation and maintenance of reliable software systems. Students should be able to demonstrate knowledge and understanding of some areas in computing that are at the forefront of the discipline. These skills will often be developed through the study of specialised optional modules in the second and final years. Students should appreciate the relative merits and limitations of different computing environments, paradigms and methodologies. Furthermore, they should be able to apply this

	knowledge to the selection of appropriate software development strategies.
• **Cognitive skills**	Will include Critical Evaluation Reflection and Communication Professional Considerations
• **Subject specific skills**	Will include Architeture Comparative programming languages Computer-based systems Databases Data structures and algorithms Middleware Multimedia Operating Systems Professionalism Programming fundamentals Software Engineering Systems analysis and design Theoretical computing Web-based computing Other areas will be included in specific final year optional modules.
• **Key Skills**	Will include Application of Number Communication Information Technology Improving Own Learning and Performance Problem Solving Working with Others

5. Structure and Regulations

Relationship Details

Module	Credits	Level	Take/Pass	Semester	Locations
CTEC1401	**30.00**	**1**	**Must Take**	**Y**	**DM**
CTEC1412	**30.00**	**1**	**Must Take**	**Y**	**DM**
CTEC1801	**30.00**	**1**	**Must Take**	**Y**	**DM**
CTEC1901	**30.00**	**1**	**Must Take**	**Y**	**DM**
CTEC2121	**30.00**	**2**	**Must Take**	**Y**	**DM**
CTEC2602	**30.00**	**2**	**Must Take**	**Y**	**DM**
CTEC2701	**30.00**	**2**	**Must Take**	**Y**	**DM**
CTEC2901	**30.00**	**2**	**Must Take**	**Y**	**DM**
SAND2802	**0.00**	**2**	**Neither**	**Y**	**DM**
CTEC3110	**15.00**	**3**	**Neither**	**Y**	**DM**
CTEC3426	**15.00**	**3**	**Neither**	**Y**	**DM**
CTEC3604	**30.00**	**3**	**Neither**	**Y**	**DM**
CTEC3901	**30.00**	**3**	**Must Take**	**Y**	**DM**
CTEC3902	**15.00**	**3**	**Must Take**	**Y**	**DM**
IMAT3404	**15.00**	**3**	**Neither**	**Y**	**DM**
IMAT3406	**15.00**	**3**	**Neither**	**Y**	**DM**
IMAT3429	**15.00**	**3**	**Neither**	**Y**	**DM**
IMAT3451	**30.00**	**3**	**Must Take**	**Y**	**DM**
IMAT3603	**30.00**	**3**	**Neither**	**Y**	**DM**
IMAT3608	**30.00**	**3**	**Neither**	**Y**	**DM**
IMAT3901	**15.00**	**3**	**Must Take**	**Y**	**DM**

Structure

Structure notes

1 notes

Course Specific Differences or Regulations

1 The requirements to progress into the sandwich placement are determined by Faculty Policy which requires that normally students must have passed a minimum of 60 credits at level 2.

Numbers at sites, including partner institutions

1

Relevant QAA Subject Benchmarking statement(s)

1 This course has been informed by the QAA Subject Benchmark Statement in Computing

6. Quality Assurance Information

QA of Workbased Learning

Liaison with Collaborative Partners

Procedures for Maintaining Standards

The Programme is managed by a programme leader together with a programme team. They are guided by the prevailing academic regulations and modular scheme handbooks produced by Registry.

An external examiner is attached to the programme who acts as a critical friend. He/She attends the assessment board and scrutinises student work and marking to ensure that standards have been maintained at an apposite level.

Each year the programme leader completes a Programme Enhancement Plan which is approved by the Programme Management Board and Faculty Academic Committee.

The student voice is heard via student representatives on the Programme Board and the Staff Student Consultative Committee. Feedback from students is gathered by end of module questionnaires and programme questionnaires.

The programme is subject to a periodic review in line with University requirements.

Course Handbook Descriptor