CO2006

Software Engineering and System Development

The Module Timetable

Dates	CMIS Week	11-13 (Mon) CW 301	14-15 (Mon) MSB LT1	15-17 (Tue) CW 301	14-16 (Wed) CW 301	16-17 (Thu) KE LT1/ CW 301	9-11 (Fri) CW 301	14-15 (Fri) CW 301	Notes
Sep 24-28	10	* no class *							Induction Week
Oct 1-5	11		L	LAB			LAB		
8-12	12	LAB		LAB			LAB		
15-19	13	CWP	R(L)	TEST		L	LAB		Sprint 1: Test: 15%
22-26	14	LAB		LAB			LAB		
29-Nov 2	15	LAB		LAB			LAB		
Nov 5-9	16	CWP		CWP		L	LAB		Sprint 2 release: 20%
12-16	17	LAB		LAB			LAB		
19-23	18	LAB		CWP		L	LAB		Sprint 3 release: 20%
26-30	19	LAB		LAB			LAB	LAB	
Dec 3-7	20	LAB		CWP		CWP	CWP		Sprint 4 release: 20%
10-14	21		R(L)		TEST				Final test: 25%

L: Lecture
R(L): Revision (Lecture)
LAB: Taught Lab Session
CWP: Coursework Preparation

CO2006 Software Engineering and System Development

Credits: *20* **Convenor:** *Dr A. Boronat, Dr J.O. Ringert* **Semester:** *1st*

Prerequisites:	*Essential: CO1003, CO1005, CO1019*	*Desirable: CO1001, CO1012*	
Assessment:	*Coursework: 100%*		
Lectures:	*6 hours*	**Class Tests:**	*2 hours*
Laboratories:	*58 hours*	**Private Study:**	*84 hours*

Subject Knowledge

Aims According to a report of the British Computer Society, only about 16% of IT projects can be considered truly successful and over 60% of them experience severe problems. The difficulties of software development led to the coining of the phrase "the software crisis" and the birth of software engineering as a discipline. However, in many companies, software is still developed in an ad-hoc way. The purpose of this module is to teach object-oriented methods for analysis, specification, design, implementation, and testing of software systems.

Learning Outcomes At the end of this course, successful students will be able to: explain the main phases in a software development process; analyse customer requirements following an agile methodology; produce object-oriented system designs, by applying design patterns and architectural styles; use UML for consistent specification of software systems and business processes; incorporate security into specifications and designs by following a flexible security specification process; and use appropriate techniques for software development and testing, including mechanisms for software reuse.

Methods Lectures, lecture notes, surgeries, recommended textbooks, worksheets, online videos, supervised laboratories, VLE discussion board, GitHub, formative feedback and web resources.

Assessment Formative coursework, online quizzes, assessed class tests and mini project.

Skills

Aims To teach students a range problem-solving skills tailored to SE, including knowledge acquisition and software modelling.

Learning Outcomes At the end of this course, successful students will be able to use short, clear summaries of technical knowledge; solve abstract and concrete problems (both routine seen, and simple unseen).

Methods Lectures, lecture notes, surgeries, recommended textbooks, worksheets, online videos, supervised laboratories, VLE discussion board, GitHub, formative feedback and web resources.

Assessment Formative coursework, online quizzes, assessed class tests and mini project.

Explanation of Prerequisites

A sound knowledge of basic algorithms, data structures and programming is required. Some knowledge of database systems of basic web application development (HTML, CSS, Javascript) is desirable.

Course Description

This module introduces students to principles and methods used to specify, design, implement and test software systems. In particular, the object-oriented paradigm will be followed, and techniques therein.

Detailed Syllabus

Introduction: Introduction to software engineering; the inherent complexity of software; examples of complex systems; basic notions and techniques for modelling software systems, build automation with Gradle, Groovy, agile principles; tasks; dependencies; repositories.

Spring MVC: architectural styles; MVC design pattern; controller; view; JSP; expression language; open and closed software architectures; vertical and horizontal prototypes; layers and partitions; coupling and cohesion; encapsulation and information hiding; subsystem decomposition; vertical and horizontal prototypes.

TDD/BDD: scenario-driven design; agile testing; TDD; BDD; user stories and use cases; JUnit; fault vs failure; the oracle problem; test model; testing activities; V model; unit testing; integration testing; system testing; acceptance testing, red/green/refactor, requirements validation and verification.

Spring Data JPA: ORM problem, JPA, entity; Hibernate; join types, fetch types; Spring Data; Query DSL; CRUD repositories;

Spring Security: threats; risks; vulnerabilities; exploits; authentication/authorization; access control with Spring Security; password storage; secure communication.

Reading List

`http://readinglists.le.ac.uk/lists/AC62C20C-59B3-3EBA-12B7-56DE98133952.html`

Resources Lecture notes, module web page, study guide, worksheets, handouts, lecture rooms with data projector, online resources, GitHub, VLE.

Module Evaluation Course questionnaires, course review.

Further Information

1 Overview

This Study Guide provides some summary information about your classes, assessment and feedback.

Victoria Park; the Medical Sciences Building is across University Road and opposite the main campus.

Note that the first lecture is on Monday October 1st at 2pm and will include a general introduction to the module.

Resources *Digital resources can be found here*

`https://campus.cs.le.ac.uk/teaching/resources/CO2006/`

Lectures Please see timetable and/or online calendar on Blackboard.

Computer Laboratories Please see timetable and/or online calendar on Blackboard.

2 Assessment

Overall Module Assessment *The final module mark is the coursework percentage mark. You require 40% to pass the module.*

The reassessment for this module in the case of failure will be 100% exam.

Coursework *Coursework consists of individual assignments (such as take home questions, class tests, mini-projects etc). Some of these are **formative** and some are **summative**. The former means that work is usually handed in and you receive feedback on the quality of your work, sometimes including a grade or mark. The latter means that the work is definitely handed in, that you will receive feedback, but always with a numerical mark that contributes to your overall module coursework mark. We sometimes refer to summative coursework as **assessed** coursework. Assessed (summative) coursework accounts for 100% of the total module mark.*

Coursework accounts for 100% of the module mark. This is computed using all assignments deemed to be **summative**.

Failure to attempt an assignment has two consequences: you will find it more difficult to understand the topic, and you will have to score higher on the remaining assignments in order to pass the module.

Normally, in order to *complete* the module assessment you need to have participated in over 50% of weighted coursework. Otherwise you will be deemed as not having completed the assessment on the module.

For an absence in a class test or a non-submission of a major assignment, a self certification alone is not sufficient for the mark to be absented. You need to provide a medical note or other appropriate evidence; your absence would then be given consideration. If no evidence is provided, you will receive a mark of 0. Overall, this missed assignment will also not count towards the 50% weighted coursework that you must participate in.

There are five weighted assignments in this module, consisting of two class tests, and a miniproject, split in three assignments. The marks stemming from each of these five assignments count towards the overall mark of the module. The table below summarises the different assignments involved in this module, including release, submission and marking deadlines where appropriate, and their assessment weight and nature. In addition, for each miniproject, there is a checkpoint deadline, usually one week before submission, which may contribute to the mark of the assignment as well.

Coursework	Type	Release	Submission	Marking	Marks	Weight
1	test	–	16/10/2018 2018	30/10/2018	100	15%
2	miniproject (I)	18/10/2018	6/11/2018	20/11/2018	100	20%
3	miniproject (II)	8/11/2018	20/11/2018	4/12/2018	100	20%
4	miniproject (III)	22/11/2018	7/12/2018	21/12/2018	100	20%
5	test	–	12/12/2018	26/12/2018	100	25%

The module convenor will arrange for one or two hours of examination revision. This will normally provide you with examples of model answers, which will be of the level, quality and detail required by examiners for the award of maximum marks. Hints will be given on the best approaches to questions and on exactly what the examiners are looking for.

3 Feedback

*The University Policy on the Return of Marked Work promises that you will receive marks and/or feedback on your coursework within **21 calendar days**. See the Student Handbook for more details. The Department of Informatics will always strive to meet this 21 calendar day deadline, and in many cases we will return marks and feedback within a shorter time period.*

When the marking has been completed, you will receive feedback, which includes the grade that you gained for the coursework and information to help you improve. The department has a standardised way to give feedback, using a short document entitled Coursework Feedback which will be made available for every assessed piece of coursework. These documents will detail how and when feedback for the coursework was or will be given. They will be available as pdf files on the module's digital resources page.