



## Foundations of Software Engineering Course Descriptor

Course Title	Foundations of Software Engineering	Faculty	Philosophy
Course code	NCHCS763	Course Leader	TBA
Credit points	15	Teaching Period	Any
FHEQ level	Level 7	Date approved	September 2020
Compulsory/ Optional	Compulsory		
Pre-requisites	None		
Co-requisites	None		

### COURSE SUMMARY

This is a group project-based course. Students will learn software engineering models, design patterns and best practices for development (e.g. testing and debugging) by working in small groups to produce a software artefact. This course is taught in the C++ programming language.

### COURSE AIMS

The aims of this course are:

- Learn about the Software Development Lifecycle
- Learn about effective design patterns (and antipatterns)
- Gain experience building larger software
- Gain experience writing tests
- Learn specific development tools (e.g. git) that will be relevant to professional careers in software development

## LEARNING OUTCOMES

On successful completion of the course, students will be able to:

### KNOWLEDGE AND UNDERSTANDING

- K1d Understand advanced aspects of the software development life cycle: development models (e.g., waterfall, spiral, agile); requirements analysis; user-centred design; software design principles and patterns; testing (functional testing, structural testing, testing strategies); code refactoring and debugging; software architecture and design; and integration and deployment.
- K2d Master practical tools, methods and techniques of highly efficient software.
- K3d Evaluate technical, social and management dimensions of software engineering based on industry standards.

### SUBJECT SPECIFIC SKILLS

- S1d Critically assess a software problem and recognise the design patterns and code organisation suitable for developing a well-tested software that solves that problems.
- S2d Become a sophisticated software engineer, familiar with codes of ethics, codes of practice and relevant industrial standards.
- S3d Design and develop original software of varying levels of complexity in C++ either individually or as part of a team.

### TRANSFERABLE AND PROFESSIONAL SKILLS

- T1d Critically review and analyse software requirements, identify limitations and propose testing and code analysis techniques to improve a project
- T2d Consistently apply an excellent level of technical proficiency in written English, using an advanced application of scholarly terminology, that demonstrates the ability to deal with complex issues both systematically and with sophistication.
- T2d Communicate effectively to peers the implementation choices and tools used to solve a problem.
- T3d Apply software engineering methods in a team-based collaborative setting.

## TEACHING AND LEARNING

Teaching and learning strategies for this course will include:

- 30 hours of full-cohort lectures
- 20 hours of lab-based tutorials
- 1 office hour per teaching week

There will be three 1-hour lectures per teaching week. Two 1-hour lab sessions will give students the opportunity to work on their assignments with the help of the course leader and teaching assistants.

Course information and supplementary materials are available on the College's Virtual Learning Environment (VLE).

Students will also attend the formal meeting, Collections, in which they will receive constructive and developmental feedback on their performance.

Students are required to attend and participate in all the formal and timetabled sessions for this course. Students are also expected to manage their directed learning and independent study in support of the course.

### EMPLOYABILITY SKILLS

- Communication Skills
- Programming skills
- Team-based project skills

### ASSESSMENT

#### FORMATIVE

Students will be formatively assessed during the course by means of set assignments. These do not count towards the end of year results but will provide students with developmental feedback. Set assignments will also amplify problem-solving skills useful for the set exercises and develop software components that form part of the students' projects.

#### SUMMATIVE

Assessment will be in two forms:

AE:	Assessment Activity	Weighting (%)	Online submission	Duration	Length
1	Set exercises	50	Yes	N/A	Code and up to 2500-word explanation
2	Project	50	Yes	N/A	Code and up to 2500-word explanation

Both the set exercises and the project will be assessed in accordance with the assessment aims set out in the Programme Specification.

### FEEDBACK

Students will receive formal feedback in a variety of ways: written (including via email correspondence); oral (within one-to-one tutorials or on an *ad hoc* basis) and indirectly through discussion during group tutorials.

Feedback is provided on summative assessment and is made available to the student either via email, the VLE or another appropriate method.

**INDICATIVE READING**

Note: Comprehensive and current reading lists for courses are produced annually in the Course Syllabus or other documentation provided to students; the indicative reading list provided below is used as part of the approval/modification process only.

**BOOKS**

MARTIN REDDY. 2011. API DESIGN FOR C++ (1ST. ED.). MORGAN KAUFMANN PUBLISHERS INC., SAN FRANCISCO, CA, USA (NOT REQUIRED)

**ELECTRONIC RESOURCES**

[C++ tutorial](#), by cplusplus.com. Last accessed August 2020 (not required)

**INDICATIVE TOPICS**

Students will study the following topics:

- Effective design patterns
- Software and test building
- Software development life cycle
- Testing and debugging in C++
- Program static analysis
- Design principles for user interfaces
- Networking

<b>Title: NCHCS763 Foundations of Software Engineering</b>					
<b>Approved by: Academic Board</b>					
<b>Location: Academic Handbook/Programme specifications and Handbooks/ Postgraduate Programme Specifications/MSc Computer Science Programme Specification/Course Descriptors</b>					
Version number	Date approved	Date published	Owner	Proposed next review date	Modification (As per AQF4) & category number
2.0	January 2022	April 2022	Dr Alexandros Koliouisis	April 2025	Category 3: Changes to Course Learning Outcomes
1.0	September 2020	September 2020	Dr Alexandros Koliouisis	April 2025	