



## MSc Software Development Individual Project Course Descriptor

|                         |   |                 |                |
|-------------------------|---|-----------------|----------------|
| Course Title            | Software Development Individual Project | Faculty         | Philosophy     |
| Course Code             | NCHCS768                                | Course Leader   | TBA            |
| Credit Points           | 60                                      | Teaching Period | Any            |
| FHEQ Level              | Level 7                                 | Date Approved   | September 2020 |
| Compulsory/<br>Optional | Compulsory                              |                 |                |
| Pre-requisites          | None                                    |                 |                |
| Co-requisites           | None                                    |                 |                |

### COURSE SUMMARY

This course provides students with the opportunity to apply state-of-the-art methods, tools and techniques in software engineering by managing a software project that solves a substantial, real-world problem. The course builds upon the variety of material being taught during the programme in earlier project-based courses (e.g. in "Foundations of Software Development").

In this course, after an initial group seminar with the course leader, students meet with an assigned supervisor to finalise the subject of their project and discuss and refine the software requirements. Once the software artefact and its documentation has been submitted, students defend it in a 40-minute presentation and demo.

### COURSE AIMS

The aims of the course are:

- Promote students' ability to design, implement and evaluate original software on a particular problem of their choice
- Promote students' ability to organise a software project from start to finish
- Promote students' ability to present clearly their ideas, design choices and evaluation methodology to their peers
- Prepare students for a wide range of careers and roles in software engineering

## LEARNING OUTCOMES

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

### KNOWLEDGE AND UNDERSTANDING

- K1d Identify, analyse, and interpret software requirements to solve a computing problem.
- K2d Demonstrate detailed critical engagement with software tools and technologies required to solve a computing problem.
- K3d Demonstrate a sophisticated understanding of software design principles, tools and techniques when developing software.

### SUBJECT SPECIFIC SKILLS

- S1d Ability to engage in a peer review process that involves critical review of software and related documentation, coupled with positive advice for improvement and innovation.
- S2d Familiarity with codes of ethics (e.g. code licencing) and codes of practice (e.g. testing) for software development underpinning the development of high quality, high integrity software systems.
- S3d Ability to recognise the individual software components required to solve a computing problem, create them or interface with them, and design their interaction.

### TRANSFERABLE AND PROFESSIONAL SKILLS

- T1d Critical review of related software, identifying key developments in a particular area, opportunities for integration, limitations and avenues for further development and innovation.
- T2d Programming skills with a range of up-to-date, well-proven software development tools and libraries.
- 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
- T4d Project leadership skills, from understanding a computing problem to proposing a software solution based on sound technical insights to encouraging others to share that vision.
- T4d Ability to convey software problems and solutions to both technical and non-technical audiences.

## TEACHING AND LEARNING

Teaching and learning strategies for this course will include:

- Group seminars (typically, 4-6 hours)
- Independent (though guided) study, research and software development
- Individual supervision, which supports both programming, writing and oral communication skills (typically, 6 hours)
- Individual written feedback
- Online discussion forum

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

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.

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

## EMPLOYABILITY SKILLS

The individual software development project cultivates the following employability skills:

- Programming skills: gather requirements, ideas and material and design and implement software using best engineering practices accordingly; conduct research and explore relevant existing software projects; solve computing problems through logical reasoning and rigorous testing
- Leadership skills: work independently, creatively and to deadlines; and engage in collaborative and constructive discussions with peers
- Communication skills: communicate findings in a clear, structured manner both orally, via presentation and demonstration of software, and in writing, via technical documentation

## ASSESSMENT

### FORMATIVE

Students will be formatively assessed during the course as they produce successive drafts of their software and related documentation. These do not count towards the end of year results, but will provide students with developmental feedback, both written and oral.

### SUMMATIVE

Assessment will be in three forms:

| AE: | Assessment Activity              | Weighting (%) | Online submission | Duration   | Length             |
|-----|----------------------------------|---------------|-------------------|------------|--------------------|
| 1   | Written Assignment               | 30%           | Yes               | N/A        | Up to 5,000 words  |
| 2   | Dissertation                     | 50%           | N/A               | N/A        | Up to 10,000 words |
| 2   | Oral Assessment and Presentation | 20%           | N/A               | 30 minutes | N/A                |

Both the project's code, the accompanying dissertation and the presentation 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 (in comments on draft material, including via email correspondence); oral (within one-to-one supervision tutorials and on an *ad hoc* basis).

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

## INDICATIVE READING

Reading is to be decided upon between student and supervisor, depending on the topic of the chosen software development project.

## INDICATIVE TOPICS

The topics covered by students will vary across projects, but typically a student will encounter (one or more times) the following topics during the project:

- Problem statement definition
- Software design and implementation
- Debugging and testing of software components
- Documentation
- Submission
- Presentation and demonstration

| <b>Title: NCHCS768 Software Development Individual Project Course Descriptor</b>   |                |                |                          |                           |   |
|--|----------------|----------------|--------------------------|---------------------------|---|
| <b>Approved by: Academic Board</b>   |                |                |                          |                           |   |
| <b>Location: Academic Handbook/Programme specifications and Handbooks/<br/>Postgraduate Programme Specifications/MSc Computer Science Programme<br/>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                |   |