

**COURSE SPECIFICATION DOCUMENT**

<b>Academic School / Department:</b>	Science, Innovation & Technology
<b>Programme:</b>	MSc Applied Computer Science (Conversion)
<b>FHEQ Level:</b>	7
<b>Course Title:</b>	Software Engineering
<b>Course Code:</b>	COMP 7107
<b>Total Hours:</b>	200
Timetabled Hours:	39
Guided Learning Hours	21
Independent Learning Hours:	140
<b>Credit</b>	20 UK CATS credits 10 ECTS credits 4 US credits

**Course Description:**

This module develops an integrated understanding of software engineering principles and structured development practices appropriate to postgraduate conversion study. It introduces core concepts in software lifecycle management, requirements analysis, system design and quality assurance, alongside the use and selection of fundamental data structures to support robust software design.

The module emphasises the disciplined application of engineering practices in the specification, development and evaluation of computer-based systems, including consideration of trade-offs relating to performance, maintainability, security and scalability.

**Prerequisites:**

None

**Aims and Objectives:**

**Aim:** To provide an integrated understanding of software engineering principles and structured development practices for the design and development of robust computer-based systems.

**Objectives:**

- Introduce the core concepts of the software development lifecycle, including requirements analysis, system design and quality assurance.
- Familiarise students with fundamental data structures and their role in supporting effective software design.
- Emphasise the application of disciplined engineering practices in the specification, development and evaluation of software systems.

- Develop awareness of key quality attributes, including performance, maintainability, security and scalability, and the trade-offs between them.

**Programme Outcomes:**

A1, A2, A4, B1, B2, B3, C1, C3, C4, D1, D4

A detailed list of the programme outcomes are found in the Programme Specification. This is located at the archive maintained by Registry and found at:

<https://www.richmond.ac.uk/programme-and-course-specifications/>

**Learning Outcomes:**

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

1. Critically evaluate software development lifecycle models and justify their application within specific project contexts.
2. Specify and design computer-based systems using appropriate modelling techniques, demonstrating awareness of architectural and quality considerations.
3. Select and apply appropriate data structures and algorithmic approaches to support efficient and maintainable software solutions.
4. Apply structured development practices, including version control, testing strategies and documentation standards, in the construction of software artefacts.
5. Analyse trade-offs relating to quality attributes such as performance, security, reliability and maintainability in the evaluation of software systems.

**Indicative Content:**

- Software development lifecycle models (waterfall, iterative, agile)
- Comparative evaluation of lifecycle approaches
- Requirements elicitation and specification
- Functional and non-functional requirements
- Architectural thinking and layered design principles
- Modularity, cohesion and coupling
- Version control workflows and collaborative development practices
- Data structures (e.g. lists, stacks, queues, hashing concepts, tree structures at conceptual level)
- Basic complexity awareness and performance trade-offs
- Design decisions informed by efficiency and scalability considerations

**Assessment:**

This course conforms to the Richmond University Standard Assessment Norms approved at Academic Board and are located at <https://www.richmond.ac.uk/university-policies/>

**Teaching Methodology:**

Teaching is delivered through lectures, seminars, design workshops, guided learning, laboratory-based activities and iterative practical tasks. Students participate in collaborative design exercises, undertake usability evaluations, and receive formative feedback that supports the development of reflective, evidence-based design practice.

**Indicative Text(s):**

- La Rocca, M. (2021). *Algorithms and Data Structures in Action*. First Edition. Shelter Island, NY: Manning Publications.
- Schutta, N. and Vega, D. (2025). *Fundamentals of Software Engineering: From Coder to Engineer*. Sebastopol, CA: O'Reilly Media, Inc.

**Journals**

*Journal of Software Engineering.*

*International Journal of Software Engineering and Computer Systems.*

See syllabus for complete reading list.

**Change Log for this CSD:**

Nature of Change	Date Approved & Approval Body (School or AB)	Change Actioned by Registry Services
Guided Learning Hours menu updated	October 2025	
Total Hours Updated	October 2025	