

Module Details	
Module Title	Software Design and Development
Module Code	COS4017-B
Academic Year	2024/5
Credits	20
School	School of Computer Science, AI and Electronics
FHEQ Level	FHEQ Level 4

Contact Hours	
Type	Hours
Lectures	11
Tutorials	4
Laboratories	33
Directed Study	148

Availability	
Occurrence	Location / Period
BDA	University of Bradford / Semester 2

Module Aims
<p>Computer software is part of everyday life across the world, providing digital technology solutions in every economic, scientific, social and health sector amongst others, including business, education, care, manufacturing, finances, travel and wellbeing, while supporting also scientific discoveries and progress.</p> <p>In this module, we will introduce core concepts of logical problem-solving with formal software analysis and design, including Object-Oriented (OO) concepts of analysis, design, development, testing, deployment, and maintenance. We will further develop theoretical understanding and practical skills for designing, developing, and evaluating software solutions that work robustly for people, society and economy. We will also introduce basic software engineering principles and software development project management and reporting knowledge and skills.</p>

Outline Syllabus

The topics of this module are broadly categorised in:

1. Introduction to the Software Design and Development lifecycle.
2. Methodologies for software requirements analysis.
3. Design techniques for usability.
4. Software testing paradigms.
5. Frameworks for software architecture.

There will be emphasis on intermediate to advanced level of (Java and other) programming language and making links between OO concepts (analysis: OOA, design: OOD, programming: OOP) using problem solving processes. The module will also introduce concepts and train your recognition of risks on data protection, legal, social, ethical and professional matters, including Generative AI influences in software solutions for problem-solving with computers.

Learning Outcomes

Outcome Number	Description
01	Discuss and apply fundamental theoretical concepts and logics concepts in problem solving for software project development.
02	Define and use fundamental principles of logic, analysis and design to express aspects of problem solving in an appropriate modelling language (e.g. UML).
03	Apply advanced Object-Oriented Analysis (OOA), Design (OOD) and Programming (OOP) concepts to construct reliable software that produces solutions for solving problems.
04	Interpret and utilise software designs expressed in an appropriate modelling language.
05	Apply individual and team skills of logical problem solving, research, project management and communication to express solutions of software design, development, testing, deployment and maintenance to case studies.
06	Demonstrate knowledge of legal, social, ethical, professional, and environmental implications of problem solving with computational solutions including in OOA, OOD, OOP, testing, deployment, and maintenance of software solutions.

Learning, Teaching and Assessment Strategy

Learning outcomes are delivered through lectures, tutorials and laboratory sessions. These contact hours aim to introduce logical problem solving, analysis, design, programming and testing concepts, introducing learners to design languages (such as UML) as well as basic software engineering methods and techniques utilised in software project development delivered individually and as part of a team.

Directed study includes reading activities, individual exercises and teamwork, and revision of concepts taught in the teaching sessions, reading and application of documentation, standards and programming examples from projects, technical reports and book sections.

Learning Outcomes 1, 2, 3, 4 are addressed in the form of individual assessments requiring consistent understanding of key OO concepts and OO notations used in logical problem solving for software analysis, design, programming, methods, tools and techniques, and business contexts.

Learning Outcomes 1, 2, 3, 4, 5, 6 are assessed through a software development team project, including programming, testing, deployment and maintenance documentation. This coursework will also assess your recognition of risks on data protection legislation, and social, ethical, professional matters.

Student re-taking the assessment during resit period will deliver this component as individual project.

.

Mode of Assessment

Type	Method	Description	Weighting
Summative	Coursework - Written	Individual project testing theoretical understanding of core computer science concepts and ability to solve problems.	50%
Summative	Coursework - Written	Group project including requirements analysis, programming, testing & documentation; Supplementary is Individual Project.	50%
Formative	Classroom test	In-class quiz and programming exercise.	N/A

Reading List

To access the reading list for this module, please visit <https://bradford.rl.talis.com/index.html>

Please note:

This module descriptor has been published in advance of the academic year to which it applies. Every effort has been made to ensure that the information is accurate at the time of publication, but minor changes may occur given the interval between publishing and commencement of teaching. Upon commencement of the module, students will receive a handbook with further detail about the module and any changes will be discussed and/or communicated at this point.