Prifysgol Wrecsam
Wrexham University

# Module specification

**When printed this becomes an uncontrolled document. Please access the Module Directory for the most up to date version by clicking on the following link: Module directory**

| | |
|---|---|
| Module Code | COM761 |
| Module Title | Advanced Software Development |
| Level | 7 |
| Credit value | 20 |
| Faculty | FACE |
| HECoS Code | 100956 |
| Cost Code | GACP |

## Programmes in which module to be offered

| Programme title | Is the module core or option for this programme |
|---|---|
| MSc Software Engineering | Core |
| MSc Software Engineering with Advanced Practice | Core |

## Pre-requisites

N/A

## Breakdown of module hours

| | |
|---|---|
| Learning and teaching hours | 10 hrs |
| Placement tutor support | 0 hrs |
| Supervised learning e.g. practical classes, workshops | 11 hrs |
| Project supervision (level 6 projects and dissertation modules only) | 0 hrs |
| **Total active learning and teaching hours** | **21** hrs |
| Placement / work based learning | 0 hrs |
| Guided independent study | 179 hrs |
| **Module duration (total hours)** | **200** hrs |

| | |
|---|---|
| **For office use only** | |
| Initial approval date | 08/11/2023 |
| With effect from date | Sept 2024 |
| Date and details of revision | |
| Version number | 1 |

## Module aims

The module mainly aims to enable the students to acquire advanced programming skills and expertise in areas such as data structures, algorithms, software design patterns and efficient resource management. Using advanced software design and architecture concepts would enable students' the ability to design scalable, maintainable and extensible software architectures using appropriate structured patterns and principles. Developing proficiency in software development methodologies, familiarizing students with advanced software development methodologies, including Agile, test-driven development (TDD), continuous integration and deployment (CI/CD) and DevOps practices.

## Module Learning Outcomes - at the end of this module, students will be able to:

| | |
|---|---|
| 1 | Apply software engineering principles, methodologies and best practices in the development of large-scale software applications. |
| 2 | Demonstrate proficiency in using a wide range of software development tools and technologies. |
| 3 | Critically analyse complex software development problems, identify appropriate solutions and evaluate their effectiveness. |

## Assessment

Indicative Assessment Tasks:
*This section outlines the type of assessment task the student will be expected to complete as part of the module. More details will be made available in the relevant academic year module handbook.*

The assessment will involve the analysis, design, and development of an application that entails tracking the development tasks of a software project for a team of developers. This application would encompass functionalities such as database backup, network communication, and analysis of recorded data. Additionally, the system would include user management features, catering to both standard users and administrators. The application would also generate reports based on multiple criteria, providing valuable insights and information.

| Assessment number | Learning Outcomes to be met | Type of assessment | Weighting (%) |
|---|---|---|---|
| 1 | 1, 2, 3 | Coursework | 100% |

## Derogations

None

## Learning and Teaching Strategies

In line with the Active Learning Framework, this module will be blended digitally with both a VLE and online community. Content will be available for students to access synchronously and asynchronously and may indicatively include first and third-party tutorials and videos, supporting files, online activities any additional content that supports their learning.

As this module progresses, the strategies will change to best support a diverse learning environment. Initially, the module will start with a heavier reliance on engaging tutor-led lectures, demonstrations, and workshops to ensure that the students get the relevant threshold concepts. As the module continues experiential and peer learning strategies will be encouraged as the students' progress with their portfolio work.

Assessment will occur throughout the module to build student confidence and self-efficacy in relation to complex software development problems.

## Indicative Syllabus Outline

Yearly content will be updated to represent the most appropriate content for current industry technologies, but a list of indicative topics could include:

- Object-oriented programming
- Functional programming concepts
- Procedural programming concepts
- Software architecture and design
- Advanced algorithms and data structures
- Database programming
- Network Programming
- Graphics Programming
- Software quality assurance
- Software security and privacy
- Emerging technologies and trends in software development
- Ethical and professional considerations
- Research and innovation in software development

## Indicative Bibliography:

Please note the essential reads and other indicative reading are subject to annual review and update.

### Essential Reads

*J. Farrell, Programming Logic & Design (Mindtap Course List), Course Technology Inc, 2023.*

### Other indicative reading

C.L. Jeffrey, *Build Your Own Programming Language: A programmer's guide to designing compilers, interpreters, and DSLs for solving modern computing problems*, Packt Publishing, 2021.

C. Althoff, *The Self-taught Programmer: The Definitive Guide to Programming Professionally*, Robinson, 2022.

CyberPunk Architects, *Computer Programming: The Bible: Learn From The Basics to Advanced of Python, C, C++, C#, HTML Coding, and Black Hat Hacking Step-by-Step IN NO TIME!,* CreateSpace Independent Publishing Platform, 2018