Module Title	Module Number	JACS Subject Code(s) and % of each subject	ASC Category(ies)
Advanced Programming	CIS6003	I320	6

Level (3 to 8)	Credits	ECTS Credit	Module Value *(1=20 credits)*	% Taught in Welsh	Module Type
6	20	10	1.0	0%	Taught

Teaching Period *(Term/Semester)*	Pre-requisites
Term 1	None

Module Leader	School(s)	Campus
Cristian Rodriguez Rivero	Cardiff School of Technologies	Llandaff

Assessment Methods

Assessment Code and Method	Duration/Length of Assessment Method	Weighting of Assessment	Threshold	Approximate Date of Submission
WRIT1 – Software Project	4000 words equivalent	100%	1	End of module

Aim(s)

This module is designed to provide the student advanced theoretical and with industry standards/practical knowledge of program and software analysis, design, and construction, contextualised by the use of appropriate tools, methodologies and techniques to create useful and usable software in industry.

Learning Outcomes

On successful completion of the module, students should be able to:

- Demonstrate fluency in contemporary programming languages, development tools and environments.
- Evaluate and demonstrate the theory and concepts of contemporary/industry standard programming and design in the software development life cycle.
- Demonstrate awareness of industry standards of professional and ethical software development, software carpentry and codemanship.

Learning and Teaching Delivery Methods

Method	Rationale	Type of Contact (scheduled/ guided independent study/placement)	Total hours
Studios	To enable core knowledge and understanding content to be delivered to the whole module cohort	Scheduled	24
Workshops	To allow exploration of all aspects of module content (knowledge, understanding, skills & other attributes) in an interactive group setting	Scheduled	24
Independent Study	To enable students to independently develop their understanding of the module concepts and to complete formative & summative assessment activity	Guided Independent Study	152
Total			200

Indicative Content

This module takes a student from being an advanced beginner to an ‘entry-level’ industry programmer, ready to start in graduate level employment.

Advanced principles of programming, programming language theory, evaluation of appropriate programming

paradigms.
Refactoring to design patterns, Design by contract, Test-driven development, pair programming
Advanced Object-Oriented Programming: Collections, iterators, domain model programming, Test-driven development, refactoring, exceptions, Design by Contract, Threads, managing memory, serialization, building GUI's, design patterns, Documentation.
Object-Oriented Theory: object-oriented design, four principles of abstraction, inheritance, encapsulation, subtype polymorphism, Open-Closed Principle, Liskov Substitution Principle, Interface Segregation Principle, Dependency Inversion Principle, Single Responsibility Principle, Design by Contract
Object-Oriented Design Practices: The Unified Process, various UML diagrams.

Specific Module Goals for OOP:
Be able to synthesize and document an object-oriented design using object-oriented analysis and design techniques.
Develop an awareness of the development processes that underlie OO development.
Learn the principles of designing and constructing Engineered Interactive Interfaces.
Develop a critical awareness of the concepts of event driven and client server application development.
Choose and implement appropriate OO solutions from problem descriptions, using a toolkit.
Critically appraise, evaluate, and implement designs using Java.
Understand current development practices and environments.
Note that these map naturally to the formal learning outcomes of the module.

Required Reading

Kerievsky, J. Refactoring to Patterns (1st ed)

Bloch, J. (2008) Effective Java (2nd ed)

Recommended Reading

Clean Code: A Handbook of Agile Software Craftsmanship 1st Edition
by Robert C. Martin

Lethbridge, T. & Laganiere, R., 2005. Object-Oriented Software Engineering. 2nd Edition. McGraw-Hill.
Cooper, J.W., 2000. Java Design Patterns. Addison Wesley.

Kernighan, B.W. and Pike, R. (1999) The Practice of Programming. Addison Wesley
Mitchell, J. C. (2002) Concepts in Programming Languages. Cambridge University Press

Dowek, G. (2009) Principles of Programming Languages. Springer

Access to Specialist Requirements

None.