MODULE TITLE	**Systems Development 2**	**CREDIT VALUE**	**15**
MODULE CODE	**ECM2429**	**MODULE CONVENER**	**Michael Saunby (Coordinator)**

DURATION: TERM	**1**	**2**	**3**
DURATION: WEEKS	**0**	**12**	**0**

Number of Students Taking Module (anticipated)	**15**

DESCRIPTION - summary of the module content

DEGREE APPRENTICESHIP STUDENTS ONLY

The purpose of this module is to extend your understanding of the fundamental concepts of systems development through programming in object oriented languages such as Python, Java and C#, computational thinking and data structures. You will analyse models of application development so that you can understand the key processes related to building functioning applications and appreciate the complexity of application development.

You will extend your skills in software design, data structures, programming, problem solving, programming logic, and fundamental software design techniques. This will include a review of traditional and contemporary software development methods including agile development. You will develop a holistic view of software engineering practice including gathering requirements, designing a solution, implementing a solution in a programming language, testing the completed application and deploying the solution to end users.

AIMS - intentions of the module

The aim of this module is to ensure that you develop your expertise in software development. In particular, you will be able to demonstrate the following competences:
Analyse business and technical requirements to select and specify appropriate technology solutions.
Design, implement, test, and debug software to meet requirements using contemporary methods including agile development.
Manage the development and assurance of software artefacts applying secure development practices to ensure system resilience.
Configure and deploy solutions to end-users.

INTENDED LEARNING OUTCOMES (ILOs) (see assessment section below for how ILOs will be assessed)

On successful completion of this module you should be able to:

Module Specific Skills and Knowledge
1. Analyse business and technical requirements and select appropriate solutions
2. Design, implement, test, and debug software to meet a requirements specification
3. Select the relevant paradigm (for example Object Oriented, Event Driven or Procedural) for a given set of business requirements
4. Write good quality code (logic) with sound syntax in at least two languages with different paradigms (e.g. object-oriented Event Driven or Procedural programming)
5. Develop moderately complex software solutions and software modifications to specified requirements
6. Design and develop user interfaces
7. Link software to databases to store and retrieve data
8. Test code and analyse results to correct errors found using unit testing.
9. Debug own code and understand structure of programs in order to identify and resolve issues
10. Identify and apply best practices and standards
11. Create data models and software designs to effectively communicate understanding of the program
12. Create analysis artefacts, such as Use Cases and/or User Stories
13. Build, manage and deploy software into enterprise environments
14. Identify and implement plans for end user training

Discipline Specific Skills and Knowledge
15. Understand basic programming concepts
16. Understand programming principles including design, code, test, correct, deploy and document from supplied specifications, using agreed standards and tools
17. Understand the stages of a software development lifecycle
18. Understand the similarities and differences (taking into account positives and negatives of both approaches) between agile and waterfall software development methodologies
19. Understand both procedural and object-oriented programming techniques
20. Be aware of the role and position of legacy systems in organisations and how new development environments interface and integrate with them.
21. Understand how teams work effectively to produce software
22. Understand software design approaches & patterns and can interpret and implement a given design (underpinning architecture how does everything fit together)

Personal and Key Transferable/Employment Skills and Knowledge
23. Communicate orally and in writing
24. Solve problems creatively
25. Think analytically and critically
26. Organise your own work
27. Work to a deadline
28. Collaborate and negotiate
29. Make decisions

SYLLABUS PLAN - summary of the structure and academic content of the module

Introduction (1 week)
Software teams
The stages of the software development lifecycle
Best practices and standards
Agile and waterfall software development methodologies
Legacy systems; role and position; integrating with legacy systems

Requirements (2 weeks)
Business and technical requirements
Creating analysis artefacts, such as use cases and/or user stories
Data modelling
Selection of solutions

Design (2 weeks)
Design communication

Software design approaches
Software patterns
Software architecture
Designing for security
User interface design

Programming (5 weeks)
Programming concepts and principles
Program structure
Software quality
Programming (procedural and object-oriented)
Selecting a programming paradigm (Object Oriented, Event Driven or Procedural)
Linking software to databases to store and retrieve data

Debugging, testing and error correction (1 week)

- Unit testing
- Integration testing
- Correcting errors

Implementation (1 week)
Deploying and managing software in enterprise environments
Rollout; data load; acceptance test
Identifying and implementing plans for end user training

LEARNING AND TEACHING

LEARNING ACTIVITIES AND TEACHING METHODS (given in hours of study time)

Scheduled Learning & Teaching Activities	22.00	Guided Independent Study	128.00	Placement / Study Abroad	0.00

DETAILS OF LEARNING ACTIVITIES AND TEACHING METHODS

Category	Hours of study time	Description
Scheduled learning and teaching activities	18	Online learning activity, including virtual workshops, synchronous and asynchronous virtual lectures and other e-learning.
Scheduled learning and teaching activities	2	Lectures
Scheduled learning and teaching activities	2	Group workshops
Guided independent study	128	Coursework, exam preparation and self-study

ASSESSMENT

FORMATIVE ASSESSMENT - for feedback and development purposes; does not count towards module grade

Form of Assessment	Size of Assessment (e.g. duration/length)	ILOs Assessed	Feedback Method
Contribution to class discussion	N/A	1-29	Verbal

SUMMATIVE ASSESSMENT (% of credit)

Coursework	60	Written Exams	40	Practical Exams	0

DETAILS OF SUMMATIVE ASSESSMENT

Form of Assessment	% of Credit	Size of Assessment (e.g. duration/length)	ILOs Assessed	Feedback Method
Individual system design exercise	60	2500 words	1, 2, 5-13, 15-18, 21, 22, 23-29	Written
Written exam	40	1 hour	1-27, 29	Written

DETAILS OF RE-ASSESSMENT (where required by referral or deferral)

Original Form of Assessment	Form of Re-assessment	ILOs Re-assessed	Time Scale for Re-assessment
Individual system design and programming exercise (60%)	Individual system design and programming exercise	1, 2, 5-13, 15-18, 21, 22, 23-29	Completed over summer with a deadline in August
Written exam (40%	Written exam (1 hour)	1-27, 29	August assessment period

RE-ASSESSMENT NOTES

Deferral – if you miss an assessment for certificated reasons judged acceptable by the Mitigation Committee, you will normally be deferred in the assessment. The mark given for a re-assessment taken as a result of deferral will not be capped and will be treated as it would be if it were your first attempt at the assessment.
Referral – if you have failed the module overall (i.e. a final overall module mark of less than 40%) you may be required to sit a referral. The mark given for a re-assessment taken as a result of referral will be capped at 40%.

RESOURCES

INDICATIVE LEARNING RESOURCES - The following list is offered as an indication of the type & level of information that you are expected to consult. Further guidance will be provided by the Module Convener

Basic reading:

ELE: ELE: http://vle.exeter.ac.uk

Web based and Electronic Resources:
John Hunt. Advanced Guide to Python 3 Programming. (1st ed. 2019), Springer, 2019.ISBN: 9783030259426, ebook ISBN: 9783030259433

Cadle, J, Ahmed, T, Cox, J., Girvan, L., Paul, A., Paul, D., Thompson, P. (2014), Developing Information Systems: Practical guidance for IT professionals (http://shop.bcs.org/display.asp?K=9781780172453#), BCS, ISBN: 9781780172453.

Reading list for this module:

Type	Author	Title	Edition	Publisher	Year	ISBN	Search
Set	Cadle, J, Ahmed, T, Cox, J., Girvan, L., Paul, A., Paul, D., Thompson, P.	Developing Information Systems: Practical guidance for IT professionals		BCS	2014	9781780172453.	[Library]
Set	Hunt, John	Advanced Guide to Python 3 Programming	1st	Springer	2019	9783030259426	[Library]
Set	Hunt, John	Advanced Guide to Python 3 Programming (ebook)	1st	Springer	2019	9783030259433	[Library]

CREDIT VALUE	15	**ECTS VALUE**	7.5

PRE-REQUISITE MODULES	None
CO-REQUISITE MODULES	None

NQF LEVEL (FHEQ)	5	**AVAILABLE AS DISTANCE LEARNING**	No
ORIGIN DATE	Thursday 06 July 2017	**LAST REVISION DATE**	Wednesday 08 February 2023

KEY WORDS SEARCH	Systems, development