



<b>Unit Title</b>	Software Design and Development (blended)
<b>FHEQ Level</b>	Level 4
<b>Unit Code</b>	CLC20105
<b>Credit Value</b>	30
<b>Unit Type</b>	Subject

Learning Hours (blended)			
Staff – Student Contact Hours		Independent Study Hours	
Classes	60	Independent study	170
Supervised access to Ravensbourne resources	30	Preparation for assessment	40
<b>Total</b>			<b>300</b>

## Unit Description

Autonomous vehicles, satellites, digital cameras, mobile phone in your hand and a digital television, one thing common in all of these is that they are controlled by software programmes.

Everything we see and do on a digital device is only possible through a programme (software). Software development is at the heart of any modern-day digital activity ranging from a contact list in your mobile phone up to controlling the airliners and satellites.

This unit is designed to give learners the knowledge and understanding required to create stand alone and web based secure programmes. The unit also covers aspects of programme design, specifications and testing methodologies.

With the help of labs and hands on practice, learners will develop an understanding of open source and vendor specific software development tools and techniques.

Students will also learn about algorithms, types of languages such as Object Oriented, Procedural and Event driven, security aspects and debugging and testing. The unit will also cover Secure Software Lifecycle and Software Security elements which maybe embedded within the programme.

On completion of this unit learners should be able to develop a suitable programme in their chosen development environment.

The Five Principles underpin the Mindsets and Skillsets Manifesto and are the foundation upon which all course curriculum frameworks and unit specifications are based. The

relevant Principles as stated below have been mapped against the Learning Outcomes relevant to each course unit and at each level (see Programme Specifications for full description of the Five Principles):

1. Cultivate / Where the individual thrives.
2. Collaborate / Where disciplines evolve.
3. Integrate / Where education engages industry.
4. Advocate / Where purpose meets practice.
5. Originate / creativity meets technology.

## Unit Indicative Content

### Industry-wide Knowledge

- Basic algorithms
- Software Development Lifecycle
- Procedural, Object Oriented and Event Driven
- Compilers and Interpreters
- Flow Charts, DFD/ERD/UML
- Architecture, DevOps
- Android Studio
- Java, C++, C#, MVC – Python
- HTML, HTML 5
- JavaScript
- WP, PHP
- Integration and Testing

### AWS Specific Knowledge Area

- Amazon Pinpoint
- CodeBuild
- CodeCommit
- CodeDeploy
- CodePipeline
- Device Farms
- SDK

### CyBOK knowledge areas

- Software Security
- Secure Software Lifecycle

### Unit Aims

1. Develop an understanding of various stages of Software Development Lifecycle, Secure Software Lifecycle and Software Security
2. Document, develop, test and evaluate a programme written in a suitable development environment
3. Apply testing and debugging techniques and reflect on the process undertaken
4. Demonstrate an understanding of techniques used in Secure Software Development

### Unit Learning Outcomes

LO 1 Research/Inspiration Demonstrate your capacity for information gathering techniques using a wide range of sources, providing visual, contextual and industry case-study research as appropriate.

Related Principle: ORIGINATE.

LO 2 Concept/Ideation

Generate first concept ideas or strategic project themes drawing upon reference to acquired research materials

Related Principle: ORIGINATE

LO 3 Development/Prototyping

Demonstrate a range of tests and solutions, informed by knowledge of the principles of the creative process.

Related Principle: INTEGRATE

LO 5 Presentation /Storytelling for Influence

Evidence effective communication of projects, whether in visual, oral or written form.

Related Principle: ADVOCATE

### Learning and Teaching Methods

This unit will be delivered using a combination of:

- Lectures / Seminars
- Online activities
- Self-directed independent study

- Peer learning, group discussion, guest speakers

#### Assessment methods and tasks

Assessment tasks	Weighting (%) <i>(one grade or multi-grade unit)</i>
Portfolio	50%
Artefact Presentation with collateral	50%

#### Indicative Assessment Criteria

Discuss iterative, agile and waterfall methods, , describe the differences between conventional SDLC and Secure Software Lifecycle (LO 1)

Document, develop, test and evaluate a programme for a given task, written in a suitable development environment such as C++, Python, PHP etc. (LO2, LO3)

Create appropriate documentation such as DFD, ERDs and Flow charts (LO5)

Apply testing and debugging techniques on your programme and discuss the results (LO3)

Apply Secure Software Development analysis tools and techniques and reflect upon its advantages and disadvantages (LO5)

#### Essential Reading list

The Cyber Security Body of Knowledge. (2019). 1st ed. The National Cyber Security Centre.

Urma, R. and Warburton, R. (2020). Real-world software development. 2nd ed. O'REILLY.

#### Recommended Reading List

Ferguson, J. (2014) BDD in Action: Behavior-driven development for the whole software lifecycle. Manning

Rod, S. (2015), Beginning Software Engineering. John Wiley & Sons, Inc