



## **Module Specification**

### **Programming for Cyber Security**

Version: 2025-26, v2.0, 04 Jun 2024

#### **Contents**

<b>Module Specification .....</b>	<b>1</b>
<b>Part 1: Information .....</b>	<b>2</b>
<b>Part 2: Description .....</b>	<b>2</b>
<b>Part 3: Teaching and learning methods .....</b>	<b>3</b>
<b>Part 4: Assessment.....</b>	<b>4</b>
<b>Part 5: Contributes towards .....</b>	<b>5</b>

## Part 1: Information

**Module title:** Programming for Cyber Security

**Module code:** UFCFGL-30-1

**Level:** Level 4

**For implementation from:** 2025-26

**UWE credit rating:** 30

**ECTS credit rating:** 15

**College:** College of Arts, Technology and Environment

**School:** CATE School of Computing and Creative Technologies

**Partner institutions:** None

**Field:** Computer Science and Creative Technologies

**Module type:** Module

**Pre-requisites:** None

**Excluded combinations:** None

**Co-requisites:** None

**Continuing professional development:** No

**Professional, statutory or regulatory body requirements:** None

## Part 2: Description

**Overview:** This module covers fundamental programming concepts, such as variables, data types, control structures, functions, arrays, I/O operations, secure coding practices and programming paradigms. Students will also learn how to switch from one to another programming language rapidly. Emphasis will be given on developing problem-solving skills, algorithmic and logical thinking, and good coding practices.

**Features:** Not applicable

**Educational aims:** This module aims to give students a solid foundational knowledge and understanding of how programming languages are used to develop solutions for variety of problems and scenarios through a practice-led approach. The module also aims to prepare students for completing challenging coding activities in their degree ahead.

**Outline syllabus:** Introduction to logical thinking and problem solving

Introduction to C and Linux

Variables and data types

Iteration (for, while, do-while)

Selection (if, else if, switch)

Arrays, Functions , and Pointers

Introduction to programming paradigms: Switch from C to Python

Lists, tuples, dictionaries, and sets

File handling operations

Secure programming (basics and advance)

Advanced programming concepts

### **Part 3: Teaching and learning methods**

**Teaching and learning methods:** Teaching and learning will be based on a triad of Theory (concept), Analysis (logical thinking and problem solving), and Practice (implementation and experimentation). Moreover, the teaching will be carried out through series of bitesize videos and interactive quiz sessions, and the learning will be achieved during the practical sessions.

This module applies the strategy of Teach Less, Learn More, Assess Less, Achieve More.

Students will be rewarded with attractive digital badges upon completion of each major milestone such as the implementation of a challenging scenario-based application based on the covered topics. This practice will continue till end of the module.

**Module Learning outcomes:** On successful completion of this module students will achieve the following learning outcomes.

**MO1** Demonstrate and apply knowledge of a range of essential programming language concepts and analyse their effectiveness for a given problem.

**MO2** Develop and express solutions algorithmically and programmatically to solve technical problems.

**MO3** Demonstrate understanding of the impact of insecure coding approaches and vulnerabilities and apply effective coding techniques to implement secure applications.

**Hours to be allocated:** 300

**Contact hours:**

Independent study/self-guided study = 228 hours

Face-to-face learning = 72 hours

Total = 0

**Reading list:** The reading list for this module can be accessed at [readinglists.uwe.ac.uk](https://rl.talis.com/3/uwe/lists/37A824D0-04F6-0070-8F82-FFCF8AA62EFE.html?lang=en-GB) via the following link <https://rl.talis.com/3/uwe/lists/37A824D0-04F6-0070-8F82-FFCF8AA62EFE.html?lang=en-GB>

## Part 4: Assessment

**Assessment strategy:** At both first sit and resit, summative assessment is achieved through a portfolio of practical tasks to evaluate the logical, problem-solving and practical skills of the students. These tasks will be based on basic and advanced concepts that students learn throughout the module. However, the focus of the tasks will be on code segments which eventually lead them towards obtaining skills related to programming for cyber security.

At main sit, students will be supported in this work through formative assessment based on the frequently given tasks such as case studies to implement a variety of programs.

**Assessment tasks:**

**Portfolio (First Sit)**

Description: The assessment consist of a portfolio of practical tasks.

Weighting: 100 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO3

**Portfolio (Resit)**

Description: The assessment consist of a portfolio of practical tasks.

Weighting: 100 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO3

**Part 5: Contributes towards**

This module contributes towards the following programmes of study:

Cyber Security and Digital Forensics [Frenchay] BSc (Hons) 2025-26

Cyber Security and Digital Forensics {Foundation} [Frenchay] BSc (Hons) 2024-25

Cyber Security and Digital Forensics {Foundation} [Frenchay] BSc (Hons) 2024-25