

WM245-18 Programming Languages for Cyber Security

24/25

Department

WMG

Level

Undergraduate Level 2

Module leader

Henry Caushi

Credit value

18

Module duration

30 weeks

Assessment

100% coursework

Study location

University of Warwick main campus, Coventry

Description

Introductory description

Programming languages are the medium through which programmers precisely describe concepts, formulate algorithms, and reason about solutions. In the course of a career, a cyber professional may work with many different languages, separately or together. If developing software, they must understand the programming models underlying different languages and make informed design choices in languages supporting multiple complementary approaches. The cyber professional is likely to need to learn new languages and programming constructs. They therefore must understand the principles underlying how programming language features are defined, composed, and implemented. The effective use of programming languages, and appreciation of their limitations, also requires a basic knowledge of programming language translation and static program analysis, as well as run-time components such as memory management.

This module enables students to develop insight into the significant differences between different programming paradigms. Object oriented programming, functional programming and event driven programming are given special emphasis. Students develop solutions under these different paradigms to embed theoretical concepts into professional practice.

Module aims

The module aims to equip students with an understanding of a number of different programming paradigms. We will consider how the choice of language or paradigm can introduce specific opportunities or vulnerabilities into software. We will also explore the factors that can influence programming language or paradigm choice, and how to keep our knowledge of software vulnerabilities up to date.

Outline syllabus

This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ.

The content of this module will be taught from a cyber security perspective.

- Object-oriented programming
- Functional programming
- Event-driven and reactive programming
- Type systems
- Program representation
- Language translation and execution
- Syntax analysis
- Compiler semantic analysis
- Code generation

Learning outcomes

By the end of the module, students should be able to:

- Identify the factors that lead to security vulnerabilities being introduced in software
- Compare and contrast the programming paradigms used to write software as well as their benefits
- Evaluate the advantages and disadvantages of different programming languages in various use cases
- Identify how choices of programming languages mitigate security vulnerabilities
- Employ sound programming practices when developing software

Indicative reading list

Aho, A. V., Lam, Monica S., Sethi, R. and Ullman, Jeffrey D., "Compilers: Principles, Techniques, and Tools", 2 Ed, Pearson (2013)

Bird, Richard, "Thinking Functionally with Haskell", Cambridge University Press (2014)

Friedman, Daniel P. and Wand, Mitchell, "Essentials of Programming Languages", 3 Revised Ed, MIT Press (2008)

Sarcar, V., "Java Design Patterns", 2nd Edition, Apress (2019)

[View reading list on Talis Aspire](#)

Subject specific skills

Compare different programming paradigms used to create software.

Reflect on how software vulnerabilities can be minimised during software creation.

Incorporate security features in small-scale programs.

Develop small-scale programs that employ the idioms of a programming paradigm in a conventional manner.

Transferable skills

Problem solving

Study

Study time

Type	Required
Lectures	18 sessions of 1 hour (10%)
Supervised practical classes	27 sessions of 1 hour (15%)
Online learning (independent)	12 sessions of 1 hour (7%)
Private study	33 hours (18%)
Assessment	90 hours (50%)
Total	180 hours

Private study description

Independent activity between workshops, programming language labs and preparation for upcoming labs.

Costs

No further costs have been identified for this module.

Assessment

You do not need to pass all assessment components to pass the module.

Assessment group A3

	Weighting	Study time
Reflective essay	40%	36 hours
A reflective essay which covers programming skills and their relevance to cyber security.		
Cyber security programming exercise	60%	54 hours
Students are tasked with developing a program which solves a given cyber security scenario, taking into account good programming practices and the need for good performance and/or security. The program is presented using a written report describing the tool and justifying the choices made during its development.		

Feedback on assessment

Written feedback for each assignment

Verbal feedback during tutorial sessions

Summative feedback on assignments and practical tests

Availability

Courses

This module is Core for:

- UWMA-H651 Undergraduate Cyber Security
 - Year 2 of H651 Cyber Security
 - Year 2 of H651 Cyber Security
 - Year 2 of H651 Cyber Security