H8SPSP: Security Principles and Secure Programming

Module Code:	H8SPSP
Long Title	Security Principles and Secure Programming APPROVED
Title	Security Principles and Secure Programming
Module Level:	LEVEL 8
EQF Level:	6
EHEA Level:	First Cycle
Credits:	10
Module Coordinator:	
Module Author:	Isabel O'Connor
Departments:	School of Computing
Specifications of the qualifications and experience required of staff	Master's and/or PhD degree in computing or cognate discipline. May have industry experience also.

Learning Outcomes

On successful completion of this module the learner will be able to:

#	Learning Outcome Description
LO1	Investigate different types of security threats and examine technologies, regulations, standards, and practices to protect individuals and organisations from cyber-attacks.
LO2	Identify and analyse common software vulnerabilities and investigate counter-measures to mitigate the threats to applications resulting from such vulnerabilities.
LO3	Evaluate, develop and implement programming solutions for securing software applications using relevant programming solutions, secure coding practices/standards, programming languages and applying secure software development lifecycle processes.
LO4	Identify, analyse and evaluate the ethical effects and impacts of design decision, the ethical issues in disclosing vulnerabilities and the ethics of thorough testing.

Dependencies

Module Recommendations

No recommendations listed

Co-requisite Modules

No Co-requisite modules listed

Entry requirements	See section 4.2 Entry procedures and criteria for the programme including procedures recognition of prior learning

H8SPSP: Security Principles and Secure Programming

Module Content & Assessment

Indicative Content
Foundational Concepts in Security Current cyber landscape. Security Goals/Properties (CIA). Authentication, authorization, access control. Concepts of trust, risk, threats, vulnerabilities and attack vectors. Security Governance, framework. Security policies, standards, guidelines
Principles of Secure Design Principles of Secure Design (least privilege, fail safe, complete mediation, open design, etc.). Tensions between security and other design goals
Secure Development Lifecycle Secure Software Development Lifecycle – include waterfall model, agile model and security. This will include threat modelling, risk assessment, incidence response and management.
Intro to Secure Coding/Defensive Programming Security support for programming languages. Type safety and its importance. Secure Coding Standards. Seven Pernicious Kingdoms
Secure Coding I: Validation of the input and its representation Input validation and data sanitization. Examples of input validation and data sanitization errors: . XSS vulnerability. SQL injection. Integer overflow. Buffer overflow.
Secure Coding II Correct Handling of exceptions and unexpected behaviour; logging & monitoring. Encapsulating structures and modules . Taking Environment into account. Using security features
Security Testing Unit testing. Code review. Static and Dynamic Analysis
Ethics in software development, testing and vulnerability disclosure. code reuse (licensing), professional responsibility, codes of ethics such as the ACM/IEEE-CS Software Engineering Code of Ethics and Professional Practice. Consequences and implications of poor or non-secure programming practices. How to disclose, to whom to disclose and when to disclose vulnerabilities. What, when and why to test – ethical implications of testing

Assessment Breakdown	%
Coursework	50.00%
End of Module Assessment	50.00%

Assessments

Full Time

Coursework

Assessment Type:	Formative Assessment	**% of total:**	Non-Marked
Assessment Date:	n/a	**Outcome addressed:**	1,2,3,4
Non-Marked:	Yes		

Assessment Description:
Ongoing tasks focused on code review, finding vulnerabilities and fixing them; discussions based on case studies, real-world examples.

Assessment Type:	Project	**% of total:**	50
Assessment Date:	n/a	**Outcome addressed:**	2,3,4
Non-Marked:	No		

Assessment Description:
Students are to develop a small application from scratch employing a secure development lifecycle model or are to be given a project that they will need to test, re-design and fix to eliminate the existent vulnerabilities.

End of Module Assessment

Assessment Type:	Terminal Exam	**% of total:**	50
Assessment Date:	End-of-Semester	**Outcome addressed:**	1,2,4
Non-Marked:	No		

Assessment Description:
Exam will consist of theoretical questions, applied theory type of questions and practical questions (e.g. code review, finding vulnerabilities in code, proposing solutions to eliminate these, etc.).

No Workplace Assessment

Reassessment Requirement

Repeat examination
Reassessment of this module will consist of a repeat examination. It is possible that there will also be a requirement to be reassessed in a coursework element.

Reassessment Description
Repeat examination Reassessment of this module will consist of a repeat examination. It is possible that there will also be a requirement to be reassessed in a coursework element.

H8SPSP: Security Principles and Secure Programming

Module Workload

Module Target Workload Hours 0 Hours

Workload: Full Time

Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Classroom & Demonstrations (hours)	24	Per Semester	2.00
Tutorial	Other hours (Practical/Tutorial)	36	Per Semester	3.00
Independent Learning	Independent learning (hours)	190	Per Semester	15.83
			Total Weekly Contact Hours	5.00

Module Resources

Recommended Book Resources

Laura Bell,Michael Brunton-Spall,Rich Smith. (2016), Agile Application Security, O'Reilly Media, p.300, [ISBN: 978-1491938843].

Matt Bishop. (2018), Computer Security, Addison-Wesley Professional, p.1440, [ISBN: 978-0-321-71233-2].

Jim Manico,August Detlefsen. (2014), Iron-Clad Java, McGraw Hill Professional, p.304, [ISBN: 978-0-07-183589-3].

This module does not have any article/paper resources

This module does not have any other resources

Discussion Note:	