MACS H6017: Secure Programming

Short Title:	Secure Programming APPROVED
Full Title:	Secure Programming

Module Code:	MACS H6017

ECTS credits:	10

NFQ Level:	9

Module Delivered in	2 programme(s)

Module Contributor:	Anthony Keane

Module Description:	This module examines the best way to develop secure applications by investigating secure coding techniques and practices.

Learning Outcomes:

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

1. Evaluate the reasons for vulnerabilities in unsecured code.
2. Research programming languages for secure features.
3. Investigate the limitations to developing secure code.
4. Design, implement and test secure programs.
5. Analyse code for vulnerabilities and research solutions.

Module Content & Assessment

Indicative Content
Overview Building Blocks of Software Security, Types of Security Vulnerabilities, Vulnerability Cycle, Types of Attacks, Hackers and Crackers or Attackers, Risk Assessment and Threat Modelling, Security Architecture, Security Principles, Secure Development Checklists
Designing Secure Architecture Introduction, Secure Architecture, Application Security, Factors Affecting Application Security, Software Engineering and System Development Life Cycle (SDLC), Different Phases of Software Development Life Cycle, Software Methodology Models, The Rules and Practices of Extreme Programming.
Secure Coding Vulnerabilities Security overview and patching, Public vulnerability databases, Secure design, principles and process, Security assessment and testing, Shell and environment, Resource exhaustion, Trust management, Buffer Overflows, Format Strings, Input Validation, Serialisation and deserialization, Input validation, Accessibility and Extensibility, Effects of superclass on subclass, Mutable class, Mutable input, Mutable output, Wrapper methods, Constants, Exceptions, SecurityManager checks,methods and non-final classes, Char and Byte array vs. String, Threads, JVM.
Secure Java and JSP Programming Introduction to Java, JVM, Java Security, Sandbox Model, Security Issues with Java, Denial-of-Service (DoS) Attack on Applet, Preventing DOS Attacks, Class File Format, Byte Code Attack, Security Policy, Specifying an additional Policy File at runtime, Policy Tool, Policy Tool: Creating a new Policy File, Best practices for developing secure Java Code
Obfuscation and packaging tools Reverse Engineering/ Decompilation, Obfuscation Tools: Jmangle, Byte Code Verifier, Class Loader, Security Manager, jarsigner - JAR Signing and Verification Tool, Signing an Applet Using RSA-Signed Certificates, Signing Tools, Getting RSA Certificates, Bundling Java Applets as JAR Files, Signing Java Applets Using Jarsigner, Signing Java Applets Using Netscape Signing Tool.
Secure Network Programming Secure Network Programming Client Server Model, Basic Web Concepts, Benefits of Secure Network Programming, Network Interface, How to Secure Sockets, Server Program, Client Program, Ports, UDP Datagram and Sockets, Internet Address, How to connect to secure websites, URL Decoder, Reading Directly from a URL, Content Handler, Cookie Policy, RMI Connector, .Net : Internet Authentication, Network Programming Best Practices, Wireless, xSEC & IPv6.

Indicative Assessment Breakdown	%
Course Work Assessment %	100.00%

Course Work Assessment %

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Written Report	Researching the reaons behind insecure applications and comparing and contrasting different programming languages.	1,2	30.00	Week 11
Lab work	Writing code segments to demonstrate secure coding techniques	3,4	35.00	Every Week
Practical/Skills Evaluation	Testing applications for vulnerabilities and fixing the code	4,5	35.00	Every Week

No Final Exam Assessment %

Indicative Reassessment Requirement
Coursework Only *This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.*

ITB reserves the right to alter the nature and timings of assessment

Indicative Module Workload & Resources

Indicative Workload: Full Time	
Frequency	**Indicative Average Weekly Learner Workload**
Every Week	30.00
Every Week	30.00
Every Week	140.00

Resources

Recommended Book Resources

Michael Howard, David LeBlanc, John Viega, *24 Deadly Sins of Software Security*, McGraw-Hill Osborne Media [ISBN: 0071626751]

Jim Manico, August Detlefsen ; technical editor, Milton Smith. 2014, *Iron-clad Java*, ; McGraw-Hill Education [ISBN: 0071835881]

Christian Cachin, Rachid Guerraoui, Lus Rodrigues 2011, *Introduction to Reliable and Secure Distributed Programming*, Springer [ISBN: 3642152597]

Theodor Richardson, Charles N. Thies., *Secure software design*, ; Jones & Bartlett Learning [ISBN: 1449626327]

Recommended Article/Paper Resources

Writing Secure Code - MSDN - Microsoft
https://msdn.microsoft.com/en-us/security/aa570401.aspx

Other Resources

Website: *Secure Coding*
http://www.infosecurity-magazine.com/sec ure-coding/

Website: *Secure Coding*
http://www.symantec.com/connect/articles /secure-coding

Website: *Secure Coding*
https://buildsecurityin.us-cert.gov/arti cles/knowledge/coding-practices

Module Delivered in

Programme Code	Programme	Semester	Delivery
BN_KMACS_R	Master of Science in Computing in Applied Cyber Security	2	Elective
BN_KMACS_M	Master of Science in Computing in Applied Cyber Security (Research)	1	Group Elective 2