NIRMA UNIVERSITY

Institute:	Institute of Technology
Name of Programme:	MTech CSE (Cyber Security)
Course Code:	3CSDE251
Course Title:	Secured Application Testing and Quality Assurance
Course Type:	(□ Core/ □ Value Added Course / √ **Department Elective** / □ Institute Elective/ □ University Elective/ □ Open Elective / □ Any other)
Year of Introduction:	2022-23

L	T	Practical Component				C
		LPW	PW	W	S	
2	-	2	-	-	-	3

Course Learning Outcomes (CLOs):

At the end of the course, the student will be able to –

1. identify various security threats in the system (BL2)
2. solve the security problems through coding (BL3)
3. evaluate the potential vulnerabilities of the system (BL5)
4. assess the security risks in the system (BL5)

Syllabus: **Total Teaching hours: 30**

Unit	Syllabus	Teaching hours
Unit-I	**Principle of Security Testing:** Introduction to secured application and its testing requirements, apply contemporary formal mathematical modeling techniques to model and analyze the security of a software system, identify project security risks & selecting risk management strategies. Principles of security testing - Confidentiality, Integrity, Authentication, Authorization, Availability, Non-repudiation. Major focus areas of the security testing- Network Security, System Software Security, Client-side Application Security and Server-side Application Security.	06
Unit-II	**Types of security testing:** Vulnerability Scanning: automated software to scan a system to detect the known vulnerability patterns, Security Scanning: identification of network and system weaknesses, reducing the defects or risks. Security scanning - manual and automated, Penetration Testing: simulation of the attack from a malicious hacker, to examine for potential vulnerabilities from a malicious hacker that attempts to hack the system, Risk Assessment: risk assessment, testing security risks, and low, medium and high risk, measures to minimize the risk. Use statistical methods to collect and analyze metrics for assessing and improving the security of a product, process, and project objectives.	06
Unit-III	**Security Auditing**: Internal inspection of applications and operating systems for security defects, line by line checking of code-static techniques, Dynamic techniques for security auditing.	06

Unit-IV	Describe and discuss security concerns designs at multiple levels of abstraction, comply with data privacy and security requirements when designing a software system, design a software solution for secure access and protection of data, use quality assurance activities and strategies that support early vulnerability detection and contribute to improving the development process.	06
Unit-V	**Ethical Hacking and Posture:** Ethical hacking, malicious hacking, identification of the security flaws in the organization system, security scanning, ethical hacking, and risk assessments to provide an overall security posture of an organization. Testing Tools: Benefits of Automation Testing, Random Testing, Bug Bashes and Beta Testing. Test Planning: Test Planning, Test Cases, Bug life cycle. Software Quality Assurance: Definition of Quality, Testing and Quality Assurance at Workplace, Test Management and Organizational Structure, Software Quality Assurance Metrics, Quality Management in IT.	06

Self-Study: The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents

Suggested Readings/ References:

1. Gary McGraw, Software Security: Building Security, Addison-Wesley
2. Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, and Nancy Mead, Software Security Engineering: A Guide for Project Managers, Addison-Wesley
3. Takanen, Ari, Jared D. Demott, Charles Miller, and Atte Kettunen. Fuzzing for software security testing and quality assurance, Artech House

Suggested List of Experiments:

Sr. No.	Title	Hours
1	Develop a program that accepts a program as input and performs check on coding ethics. Take any programming language of your choice.	04
2	Demonstration of various source code analysis tools.	04
3	Demonstration of various IDEs to perform the software testing using preferences option.	04
4	Use open-source tools to perform risk assessment like Open-Source Risk Engine, SimpleRisk, Eramba, RA Risk Coverage, PTA Professional, OpenVAS	06
5	Demonstration of Project Risk Manager – Cloud based version	06
6	Perform security testing of application using tools like Netsparker, ImmuniWeb, Vega, Wapiti	06