

Integral University Lucknow
Study & Evaluation Scheme
M. Tech. (Evening) Computer Science & Engg.

Semester -II

S. No.	Subject Code	Category	Subject	Periods				Evaluation Scheme				Subject Total
				L	T	P	C	C A			Exam.	
								CT	TA	Total	ESE	
1	CSE 510	DC	Cryptography & Networks Security	3	1	0	4	40	20	60	40	100
2	CSE 511	DC	Software Testing and Quality Management	3	1	0	4	40	20	60	40	100
3	CSE 512	DC	Advanced Software Engineering Lab	0	0	3	2	30	30	60	40	100
Total				6	2	3	10	80	60	140	160	300

L-Lecture **T**-Tutorial **P**-Practical **C**-Credits **CT**-Class Test

TA-Teacher Assessment

CA - Continuous Assessment

Continuous Assessment = Class Test + Teacher Assessment

Subject Total = Continuous Assessment(CA) + End Semester Examination (ESE)

DC- Departmental Core

DE- Departmental Elective

Integral University, Lucknow
Department of Computer Science & Engineering
M.Tech. (Evening) CSE 1st Year / 2nd Semester
Subject Name: Cryptography And Network Security, Subject Code: CSE- 510
SYLLABUS REVISED-2015
w.e.f. July - 2016

L T P C
3 1 0 4

UNIT 1

Introduction to OSI Security Architecture: Conventional Encryption: Conventional Encryption Model, Classical Encryption Techniques - Substitution Ciphers: Transpositions Ciphers: Cryptanalysis, Staganography; Modern Block Ciphers- Block Ciphers Principles: Stream & Block Ciphers, Fiestal Cipher, Shannon's Theory of Confusion and Diffusion, Data Encryption Standards (DES): DES Encryption and Decryption, Strength of DES, , Block Cipher Modes of Operation [8]

UNIT 2

Triple DES: Double DES, TDES with Two Keys, TDES with Three Keys; Confidentiality Using Conventional Encryption: Placement of Encryption Function, Traffic Confidentiality, Key Distribution, Random Number Generation ,Introduction to Graph, Ring and Field, Prime and Relative Prime Numbers, Modular Arithmetic,. Principles of Public Key Cryptosystems: RSA Algorithm: Key Management, Diffie-Heilman Key Exchange Algorithm, And Introductory Idea of Elliptic Curve Cryptography. [8]

UNIT 3

Message Authentication & Hash Functions: Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions: Requirement for a Hash Function, Simple Hash Functions, Block Chaining Techniques; Security of Hash Function & MACS, MD5 Message Digest Algorithm, Secure Hash Algorithm (SHA-1).. Digital Signatures: Requirements, Direct & Arbitrated Digital Signature; Authentication Protocols: Mutual & One way Authentication; Digital Signature Standard (DSS) [7]

UNIT 4

Network Security Application- Authentication Applications: Kerberos Version 4 & Difference between Kerberos v4 & v5, Kerberos Realms; X.509 Authentication Service: Electronic Mail Security - Pretty Good Privacy (PGP): IP Security: Architecture, Authentication Header, Encapsulating Security Payloads. Web Security: Secure Socket Layer & Transport Layer Security, Secure Electronic Transaction (SET); System Security: Intruders, Viruses and Related Threats: Malicious Programs, The Nature of Viruses, Types of Viruses, Macro Viruses, Email Viruses; Firewall: Firewall Design Principles, Trusted Systems. [9]

UNIT 5

Cyber Laws: Cyber laws for Cyberspace- Legal Identity and Private International Laws in Cyberspace. IT Act 2000. IT Act 2000 in reference to Email, E-Commerce and Issues of Privacy. E-Agreements and the Web Surfing, Cyber Pirates - Copyright, Digital Content right. Steps to Protect the Contents on WWW, Software Patents, Domain Name System and Trademarks, Cyber Trademarks Laws. IT Act and Issues of Copyright, Patent and Trademark. Crimes- Cyber Crimes and Future Imperfect, Strategy to Combat Cyber Crimes, IT Act 2000 and Cyber Crimes. Case Study of Sensor Network Security. [8]

REFERENCES:

1. William Stallings, "Cryptography and Network Security: Principles and Practice" Prentice Hall, New Jersey.
2. Johannes. A. Buchmann, "Introduction to cryptography", Springer Verlag. Bruce Schiener, "Applied Cryptography".
3. Sharma, Vakul, "Handbook of cyber Laws", Macmillan India Ltd, 2002.
4. Kernal Texpalan, " Communication network Management:", PHI, 1992.

Integral University, Lucknow
Department of Computer Science & Engineering
M.Tech. (Evening) CSE 1st Year / 2nd Semester
Subject Name: Software Testing & Quality Management, Subject Code: CSE- 511
SYLLABUS REVISED-2015
w.e.f. July - 2016

L T P C
3 1 0 4

UNIT -I

Introduction to Software Testing: Evolution, Myths, Facts, Goals, Psychology, Models, Principles , Axiom of Testing, Study of Bugs: Classification, Priority, Severity and their tracking .**Software Testing:** Terminology, Life cycle, Methodology, Types of Testing, Test planning: Test Plan Specification, Leveled Test Plan, Development of Test Plan ,Master Test plan ,Phase Wise Test Plan, Test management, Software Testing Guidelines, Defect Management, Analyzing & Reporting Test. [7]

UNIT -II

Testing Technique: Static Testing: Inspection, Structured Walkthrough, Technical reviews, Automated Techniques ,Black box testing, Types of Black box Testing: Requirement based Testing, Positive & Negative Testing, Boundary Value Analysis, Compatibility Testing, Domain Testing, Graph Based Testing, Robustness Testing , Syntax Testing, Finite State Testing, Cause-Effect Graphing Based Testing. White Box Testing, Types of White box Testing: Basis Path Testing, Control Structure Testing, Mutation Testing, and Gray Box Testing. Software Testability, Software Test Automation, Test Metrics and its Measurements. [9]

UNIT -III

Software Testing Strategies: Model for Software Testing, Unit Testing, Integration, System & Acceptance Testing: Alpha Testing, Beta Testing, Stress testing, Load testing, Reliability Testing, Scalability Testing . Performance Testing, Regression Testing, Ad-hoc Testing, Usability and Accessibility Testing, Object Oriented Testing: Object-Oriented Testing Model, Object-oriented Software Test Strategy. [8]

UNIT -IV

Introduction to Software Quality: Concepts of quality, perspectives and expectations, Quality Framework, **Quality engineering:** Activity and process, Quality planning, Quality assessment and improvement. **Quality assurance:** Classification, Q.A activities, Q.A. Techniques, Defect prevention and process improvement, Software Inspection, Formal Verification, Introduction to Software Reliability Engineering, Software Quality Measurement & Metrics. [7]

UNIT -V

Quality Models: McCall's model, Bohem's model, Dromey's model, FURPS Model,ISO-9126 Model, Cost Of Quality, Software Quality Factors, Quality Control , CMMI-Framework : Process Area Components, Capability & Maturity Levels, Relationship Among Process Areas. [9]

REFERENCES:

1. Software Testing : K.Mustafa,R.A. Khan ,Narosa
2. Software Testing : Srinivasan Desikan,Pearson
3. Software Testing : Naresh Chauhan , Oxford
4. Software Quality Engineering : Jeff Tian ,Wiley
5. Software Testing Fundamentals: Marnie L.Hutcheson,Wiley
6. Software Testing : Ron Patton, Pearson

Integral University, Lucknow
Department of Computer Science & Engineering
M.Tech. (Evening) CSE 1st Year / 2nd Semester
Subject Name: Advanced Software Engineering Lab, Subject Code: CSE- 512
SYLLABUS REVISED-2015
w.e.f. July - 2016

L T P C
0 0 3 2

1. Use a good SRS-document (available in Project Data Bank) and review this from TESTABILITY perspective. Identify “un-testable requirements” in SRS and suggest corrective measures – for making the requirements testable.
2. Research “Requirements Management Tools” currently available in the market (including OPEN-SOURCE TOOLS as well) and provide comparative presentation – with recommendations for your lab – if you were given a chance to make decision on tool selection, what would you deploy and why?
3. Identify a real-life application and develop a test-plan, for this application.
4. Research currently available OPEN-SOURCE Testing Tools and identify one of them (that you believe is most suitable), for deployment in the lab. Make a presentation for End-to-End Deployment, for the TEST-Tool of your choice.
5. Write application programs in C-Language (design multiple modules) – using various constructs of C-Language and perform “WHITE-BOX TESTING” of: (a.) Individual Modules (b.) Interfaces.

During the course of White-Box Testing, ensure (and provide evidence of) the following:

- i. All independent paths within a module have been covered at least once
 - ii. Cover all logical decisions on their true and false sides
 - iii. Execute all loops at their boundaries and within their operational bounds
 - iv. Exercise internal data structures to ensure their validity
6. Identify a good SRS-document (currently available in Project Data Bank) and prepare UAT Test-Cases on the basis of the SRS-document.
 - 7.
 8. Develop Test-Cases for End-to-End, Black-Box Testing of the application program of exercise(5.). Execute the test-cases and retain detailed test-log and other artifacts of testing. Present Test-Summary report and answer queries (with reference to test artifacts).
 9. Study any Test Management Tool (like “TEST DIRECTOR” or any other tool of your choice) in detail and provide a presentation/evaluation demo on this tool in the lab.
 10. Research and Identify a Test Tool for Automating Test Scripts. Compare the same with WinRunner and provide your objective assessment comments.
 11. Take a mini-project and handle the same using V-Process Model of SDLC. During life cycle of this mini-project, create various Testing Documents and the Final Test Report, as deliverables.