Gates Institute of Technology

(UGC Autonomous)

Gooty, Ananthapuramu Dist. – 515 002 (AP) India

==========

B. Tech (Regular-Full time)

(Effective for the students admitted into I B.Tech from the Academic B.Tech 2023-24 onwards)

CSE – CYBER SECURITY

COURSE STRUCTURE

&

SYLLABUS

III B.Tech I Semester (CSE-Cyber Security)

S.No.	Course Code	Title	L	T	P	Credits
1	23CST22	Cloud Computing	3	0	0	3
2	23CYT02	Cyber Security	3	0	0	3
3	23CST10	Automata Theory & Compiler Design	3	0	0	3
4	23CST12	Introduction to Quantum Technologies and Applications	3	0	0	3
5	23CST07 23ECT16 23ECT12 23CST08	**Professional Elective-I** 1. Software Engineering 2. Wireless Sensor Networks 3. Microprocessors & Microcontrollers 4. Artificial Intelligence	3	0	0	3
6		**Open Elective- I**	3	0	0	3
7	23CSP16	Cloud Computing Lab	0	0	3	1.5
8	23CYP02	Cyber Security Lab	0	0	3	1.5
9	23CSP12	**Skill Enhancement course-III** Full Stack Development-2	0	1	2	2
10	23ECP09	Tinkering Lab	0	0	2	1
11	23CSCSI	Evaluation of Community Service Internship	-	-	-	2
Total			**15**	**1**	**10**	**26**

12 week MOOC Swayam/NPTEL course recommended by the BoS

S.No	Course Name	Link
1	Artificial Intelligence	https://onlinecourses.nptel.ac.in/noc25_cs07/preview

Open Elective – I

S.No.	Course Code	Course Name	Offered by the Dept.
1	23CET12	Green Buildings	CIVIL
2	23CET13	Construction Technology and Management	
3	23EET13	Electrical Safety Practices and Standards	EEE
4	23MET14	Sustainable Energy Technologies	ME
5	23ECT17	Electronic Circuits	ECE
6	23BST19	Mathematics for Machine Learning and AI	Mathematics
7	23BST20	Materials Characterization Techniques	Physics
8	23BST21	Chemistry of Energy Systems	Chemistry
9	23BST22	English for Competitive Examinations	Humanities
10	23BST23	Entrepreneurship and New Venture Creation	

Note:

1. A student is permitted to register for Honours or a Minor in IV semester after the results of III Semester are declared and students may be allowed to take maximum two subjects per semester pertaining to their Minor from V Semester onwards.
2. A student shall not be permitted to take courses as Open Electives/Minor/Honours with content substantially equivalent to the courses pursued in the student's primary major.
3. A student is permitted to select a Minor program only if the institution is already offering a Major degree program in that discipline.

III B. Tech I Semester

23CST22	**CLOUD COMPUTING**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Course Objectives:

- To explain the evolving computer model called cloud computing.
- To introduce the various selves of services that can be achieved by cloud.
- To describe the security aspects in cloud.

Course Outcomes(CO):

After completion of the course, students will be able into

- Ability to create cloud computing environment.
- Ability to design applications for Cloude environment.
- Design & develop backup strategies for cloud databased on features.
- Use and Examine different cloud computing services.
- Apply different cloud programming model as per need.

UNIT-I: Basics of Cloud computing Lecture 8 Hrs Introduction to cloud computing :Introduction, Characteristics of cloud computing, Cloud Models, Cloud Services Examples, Cloud Based services and applications Cloud concept and Technologies :Virtualization, Load balancing, Scalability and Elasticity, Deployment, Replication, Monitoring, Software defined, Network function virtualization, Map Reduce, Identity and Acce ss Management, services level Agreements, Billing.

Cloud Services and Platforms: Compute Services, Storage Services, Database Services, Application services, Content delivery services, Analytics Services Deployment and Management Services, Identity and Access Management services, Open Source Private Cloud software.

UNIT-II: Hato and Python Lecture 9 Hrs

Hardtop Map Reduce: Apache Hardtop, Hardtop Map Reduce Job Execution, Hardtop Schedulers, Hoodoop Cluster setup. Cloud Application Design: Reference Architecture for Cloud Applications, Cloud Application Design Met homologies, Data Storage Approaches. Python Basics: Introduction, Installing Python, Python data Types Data Structures, Control flow, Function, Modules, Packages, File handling, Date/Time Operations, Classes.

UNIT-III: Python for Cloud computing Lecture 8 Hrs Python for Cloud: Python for Amazon webservices, Python for Google Cloud Platform, Python for windows Azure, Python for Map Reduce, Python package self- interest, Python web Application Framework, Designing a Restful web API.

Cloud Application Development in Python: Design Approaches, Image Processing APP, Document Storage App, Map Reduce App, Social Media Analytics App.

UNIT-IV: Bigdata, multimedia and Tuning Lecture 8 Hrs Big Data Analytics: Introduction, Clustering Big Data, Classification of Bigdata Recommendation of Systems.

Multimedia Cloud: Introduction, Case Study: Live video Streaming App, Streaming Protocols, case Study: Video Transco ding App.

Cloud Application Benchmarkingand Tuning:Introduction, Workload Characteristics, Application Per formance Metrics, Design Considerationsfor a Benchmarking Methodology, Benchmarking Tools, Deployment Prototyping, Load Testing&Bottleneck Detectioncase Study, Hardtopbenchmarkingcase Study.

UNIT-V:Applicationsand Issuesin Cloud Lecture 9 Hrs Cloud Security: Introduction, CSA Cloud Security Architecture, Authentication, Authorization, Identity Access Management, Data Security, Key Management, Auditing.
Cloudfor Industry, Healthcare&Education:Cloud Computingfor Healthcare, Cloud computingfor Energy Systems, Cloud Computingfor Transportation Systems, Cloud Computingfor Manufa cturing Industry, Cloudcomputing for Education.
Migratingintoa Cloud:Introduction, Broad Approach est. migratingintothecloud, theseven–stepmodel ofmigrationintoacloud.

Organizationarea dinesand Change Management in The Cloud Age:Introduction, Basic concepts of Organizational Readin ess, Driversforchanges:Aframeworktocomprehendthecompetitiveenvironment, commonchangemanag ementmodels, changemanagementmaturatemodels, Organizationareadines self–assessment.

Legal Issues Cloud Computing:Introduction, Data Privacyandsecurity Issues, cloudcontractingmodels, Jurisdictionalissuesr aisedbyvirtualizationanddatalocation, commercialandbusinessconsiderations, Special Topics.

Textbooks:

1. Cloud computing A hands-on Approach‖By Arshdeep Bahga, Vijay Marinetti, Universities Press, 2016
2. Cloud Computing Principlesand Paradigms:By Raj Kumar Buyya, James Bromberg, And razes Goscinski, Wiley, 2016

Reference Books:

1. Mastering Cloud Computingby Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, TMH
2. Cloudcomputing A Hands-On Approachby Arshdeep Bahgaand Vijay Madisetti.
3. Cloud Computing:A Practical Approach, Anthony Svelte, Toby J. Velte, Robert Elsenpeter, Tata Mc Graw Hill, rip 2011.
4. Enterprise Cloud Computing, Gautam Sheriff, Cambridge University Press, 2010.
5. Cloud Application Architectures:Building Applicationsand Infrastructure in the Cloud, George Reese, O_Reilly, SPD, rp 2011.
6. Essentialsof Cloud Computingby K. Chandrasekaran. CRC Press.

Online Learning Resources:

Cloudcomputing – Course(nptel. ac. in)

III B.Tech I Semester

		L	T	P	C
23CYT02	**CYBER SECURITY**	**3**	**0**	**0**	**3**

Course Objectives:

1. Develop a foundational comprehension of cybersecurity concepts, encompassing threats, vulnerabilities, and protective strategies.
2. Identify and categorize common cyber threats, understand their propagation, and implement effective countermeasures.
3. Explore techniques for ensuring data integrity, authentication, and data availability, while comprehending cryptographic controls.
4. Develop skills to respond to cybersecurity incidents, execute disaster recovery plans, and enhance system availability.
5. Analyse the ethical dimensions of cybersecurity, understand professional responsibilities, and uphold ethical standards in the field.

Course Outcomes:

1. Demonstrate various cyber threats and vulnerabilities, understanding their potential impact on digital assets.
2. Implement proactive measures to mitigate cyber threats and protect against common attack vectors.
3. Apply cryptographic techniques to ensure data integrity, authenticity, and confidentiality.
4. Develop incident response plans and disaster recovery strategies to minimize the impact of cybersecurity incidents.
5. Understand to ethical principles and professional responsibilities while making informed decisions in the realm of cybersecurity

UNIT-I : Cybersecurity Essentials and Cube 9 Hrs

The Cybersecurity World, Cyber Criminals versus Cybersecurity Specialists, Common Threats, Spreading Cybersecurity Threats, The Three Dimensions of the Cybersecurity Cube, CIA Triad, States of Data, Cybersecurity Countermeasures, IT Security Management Framework.

UNIT-II : Cybersecurity Threats, Vulnerabilities, Attacks and Protecting Secrets9Hrs Introduction, Governance, Managing Cloud Security Risk, Compliance, Legal Issues in Cloud, Audit, CSA Tools.

UNIT-III: Data Integrity 9 Hrs

Types of Data Integrity Controls, Digital Signatures, Certificates, Database Integrity Enforcement.

UNIT-IV: Data Availability and Recovery 9 Hrs High Availability, Measures to Improve Availability, Incident Response, Disaster Recovery.

UNIT-V: Protecting a Cybersecurity Domain 9 Hrs

Defending Systems and Devices, Server Hardening, Network Hardening, Physical and Environmental Security, Cybersecurity Domains, Ethics of Working in Cybersecurity.

Text Books:

1. CISSP (ISC)2 Certified Information Systems Security Professional Official Study Guide, Mike Chappell, James Michael Stewart and Darrel Gibson, 9th Edition
2. Cybersecurity: The Beginner's Guide, Dr.ErdalOzkaya, Packt Publishing Limited, 2019.

Reference Books:

1. Cybersecurity Essentials, Charles J. Brooks, Christopher Grow, Philip Craig and Donald Short, 1st edition, Say box.
2. Network Security Essentials, William Stallings, 6th edition, Pearson Education, 2018.

III B.Tech I Semester

23CST10	**AUTOMATA THEORY & COMPILER DESIGN**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Course Objectives:

1. Able to understand the concept of abstract machines, construct FA, Regular Expressions for the regular languages and equivalent FSMs.
2. Able to construct pushdown automata equivalent to Context free Grammars, construct Turing Machines and understand undecidability.
3. Emphasize the concepts learnt in phases of compiler, lexical analyser and Top-down parser.
4. Able to understand the concepts of Bottom-up parser, Intermediate Code Generation.
5. Able to understand the concepts of Code optimizer and Code Generation.

Course Outcomes:

1. Demonstrate knowledge on Automata Theory, Regular Expression and Analyze and Design of finite automata, and prove equivalence of various finite automata.
2. Demonstrate knowledge on context free grammar, Analyze and design of PDA and TM.
3. Understand the basic concept of compiler design, and its different phases which will be helpful to construct new tools like LEX, YACC, etc.
4. Ability to implement semantic rules into a parser that performs attribution while parsing and apply error detection and correction methods.
5. Apply the code optimization techniques to improve the space and time complexity of programs while programming and Ability to design a compiler.

Unit-I: Introduction to Automata and Regular Expressions 12 Hrs

Introduction, Alphabets, Strings and Languages, Chomsky Hierarchy, Automata and Grammars, Regular Grammar and Language, Finite Automata, Deterministic finite Automata (DFA), Nondeterministic finite Automata (NFA), Equivalence of NFA and DFA, Minimization of Finite Automata, Regular Expressions, Converting Regular Grammar and Expression into Finite Automata, Pumping lemma for regular sets, Closure properties of regular sets (Without proof).

UNIT-II: Context Free Grammars and Pushdown Automata 12 Hrs

Context Free Language, Context Free Grammar, Derivation and Parse tree, Ambiguity, Simplification of CFG‘s, Chomsky Normal Form, Greibach Normal Form, Push Down Automat (PDA), Design of PDA, Equivalence of PDA and CFL/CFG

UNIT-III: Turing Machines and Introduction to Compilers 12 Hrs

Turing Machine, TM Model, Language acceptance, Design of Turing Machine, Compilers, Phases of Compiler, The role of Lexical Analyzer, Input Buffering.

UNIT-IV: Parsers and Intermediate Code Generation 12 Hrs

Parser, Top-Down parsers: Recursive Descent Parsers, Predictive Parsers
Bottom–up Parsers: Shift-Reduce Parsing, LR parsers, Intermediate Code Generation: Three address codes.

UNIT-V: Code Optimization and Code Generation 12 Hrs

Code Optimization: Peephole optimization, Basic blocks and flow graphs, DAG, Principles of Source Code Optimization, Code Generation: Issues in Design of Code Generation, Simple Code Generator.

Text Books:

1. Introduction to Automata theory languages and Computation, Hopcroft H.E. and Ullman Jeffrey.D, 3/e, 2006, Pearson Education, New Delhi, India.

2. Mishra K L P and Chandrasekaran N, ―Theory of Computer Science - Automata, Languages and Computation‖, 2/e, 2007, PHI, New Delhi, India.
3. Compilers: Principles, Techniques, and Tools, Updated 2e July 2023 Alfred V. Aho , Monica S. Lam, Ravi Sethi , Jeffrey D. Ullman , Sorav Bansal

Reference Books:

1. Introduction to Languages and Theory of Computation, John C Martin, 1/e, 2009, Tata McGraw Hill Education, Hyderabad, India.
2. Introduction to Theory of Computation, Sipser, 2/e, 2005, Thomson, Australia.
3. Compiler Construction: Principles And Practice, Kenneth C. Louden, Thomson/ Delmar Cengage Learning, 2006.
4. Lex &yacc, Doug Brown, John Levine and Tony Mason, 2 nd Edition, O‘reilly Media
5. Engineering a compiler, Keith Cooper and Linda Torczon, 2 nd Edition, Morgan Kaufmann, 2011.

		L	T	P	C
23CST12	**INTRODUCTION TO QUANTUM TECHNOLOGIES AND APPLICATIONS** (Qualitative Treatment)	**3**	**0**	**0**	**3**

Course Objectives (COBJ):

- Introduce fundamental quantum concepts like superposition and entanglement.
- Understand theoretical structure of qubits and quantum information.
- Explore conceptual challenges in building quantum computers.
- Explain principles of quantum communication and computing.
- Examine real-world applications and the future of quantum technologies.

Course Outcomes (CO):

- Explain core quantum principles in a non-mathematical manner.
- Compare classical and quantum information systems.
- Identify theoretical issues in building quantum computers.
- Discuss quantum communication and computing concepts.
- Recognize applications, industry trends, and career paths in quantum technology.

Unit 1: Introduction to Quantum Theory and Technologies

The transition from classical to quantum physics, Fundamental principles explained conceptually: Superposition, Entanglement, Uncertainty Principle, Wave-particle duality, Classical vs Quantum mechanics – theoretical comparison,Quantum states and measurement: nature of observation,Overview of quantum systems: electrons, photons, atoms,The concept of quantization: discrete energy levels,Why quantum? Strategic, scientific, and technological significance,A snapshot of quantum technologies: Computing, Communication, and Sensing,National and global quantum missions: India‘s Quantum Mission, EU, USA, China

Unit 2: Theoretical Structure of Quantum Information Systems

What is a qubit? Conceptual understanding using spin and polarization, Comparison: classical bits vs quantum bits, Quantum systems: trapped ions, superconducting circuits, photons (non-engineering view),Quantum coherence and decoherence – intuitive explanation, Theoretical concepts: Hilbert spaces, quantum states, operators – only interpreted in abstract,The role of entanglement and non-locality in systems, Quantum information vs classical information: principles and differences,Philosophical implications: randomness, determinism, and observer role

Unit 3: Building a Quantum Computer – Theoretical Challenges and Requirements

What is required to build a quantum computer (conceptual overview)?,Fragility of quantum systems: decoherence, noise, and control, Conditions for a functional quantum system: Isolation, Error management, Scalability, Stability, Theoretical barriers:

Why maintaining entanglement is difficult,Error correction as a theoretical necessity, Quantum hardware platforms (brief conceptual comparison),Superconducting circuits, Trapped ions, Photonics, Visionvs reality: what‘s working and what remains elusive,The role of quantum software in managing theoretical complexities

Unit 4: Quantum Communication and Computing – Theoretical Perspective

Quantum vs Classical Information, Basics of Quantum Communication, Quantum Key Distribution (QKD),Role of Entanglement in Communication,The Idea of the Quantum Internet – Secure Global Networking,Introduction to Quantum Computing,Quantum Parallelism (Many

States at Once),Classical vs Quantum Gates, Challenges: Decoherence and Error Correction,Real-World Importance and Future Potential

Unit 5: Applications, Use Cases, and the Quantum Future

Real-world application domains: Healthcare (drug discovery),Material science, Logistics and optimization, Quantum sensing and precision timing, Industrial case studies: IBM, Google, Microsoft, PsiQuantum,Ethical, societal, and policy considerations, Challenges to adoption: cost, skills, standardization,Emerging careers in quantum: roles, skillsets, and preparation pathways,Educational and research landscape – India's opportunity in the global quantum race

Textbooks:

1. Michael A. Nielsen, Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 10th Anniversary Edition, 2010.
2. Eleanor Rieffel and Wolfgang Polak, *Quantum Computing: A Gentle Introduction*, MIT Press, 2011.
3. Chris Bernhardt, *Quantum Computing for Everyone*, MIT Press, 2019.

Reference Books:

1. David McMahon, *Quantum Computing Explained*, Wiley, 2008.
2. Phillip Kaye, Raymond Laflamme, Michele Mosca, *An Introduction to Quantum Computing*, Oxford University Press, 2007.
3. Scott Aaronson, *Quantum Computing Since Democritus*, Cambridge University Press, 2013.
4. **Alastair I.M. Rae**, *Quantum Physics: A Beginner's Guide*, Oneworld Publications, Revised Edition, 2005.
5. **Eleanor G. Rieffel, Wolfgang H. Polak**, *Quantum Computing: A Gentle Introduction*, MIT Press, 2011.
6. **Leonard Susskind, Art Friedman**, *Quantum Mechanics: The Theoretical Minimum*, Basic Books, 2014.
7. **Bruce Rosenblum, Fred Kuttner**, *Quantum Enigma: Physics Encounters Consciousness*, Oxford University Press, 2nd Edition, 2011.
8. **GiulianoBenenti, GiulioCasati, GiulianoStrini**, *Principles of Quantum Computation and Information, Volume I: Basic Concepts*, World Scientific Publishing, 2004.
9. **K.B. Whaley et al.**, *Quantum Technologies and Industrial Applications: European Roadmap and Strategy Document*, Quantum Flagship, European Commission, 2020.
10. **Department of Science & Technology (DST), Government of India**, *National Mission on Quantum Technologies & Applications – Official Reports and Whitepapers*, MeitY/DST Publications, 2020 onward.

Online Learning Resources:

- IBM Quantum Experience and Qiskit Tutorials
- Coursera – Quantum Mechanics and Quantum Computation by UC Berkeley
- edX – The Quantum Internet and Quantum Computers
- YouTube – Quantum Computing for the Determined by Michael Nielsen
- Qiskit Textbook – IBM Quantum

III B.Tech I Semester

23CST07	**SOFTWARE ENGINEERING (Professional Elective-1)**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Course Objectives:

The objectives of this course are to introduce

- Software life cycle models, Software requirements and SRS document.
- Project Planning, quality control and ensuring good quality software.
- Software Testing strategies, use of CASE tools, Implementation issues, validation & verification procedures.

Course Outcomes:

After completion of the course, students will be able to

1. Perform various life cycle activities like Analysis, Design, Implementation, Testing and Maintenance (L3)
2. Analyse various software engineering models and apply methods for design and development of software projects. (L4)
3. Develop system designs using appropriate techniques. (L3)
4. Understand various testing techniques for a software project. (L2)
5. Apply standards, CASE tools and techniques for engineering software projects (L3)

UNIT- I:

Introduction: Evolution, Software development projects, Exploratory style of software developments, Emergence of software engineering, Notable changes in software development practices, Computer system engineering.

Software Life Cycle Models: Basic concepts, Waterfall model and its extensions, Rapid application development, Agile development model, Spiral model.

UNIT- II:

Software Project Management: Software project management complexities, Responsibilities of a software project manager, Metrics for project size estimation, Project estimation techniques, Empirical Estimation techniques, COCOMO, Halstead's software science, risk management.

Requirements Analysis And Specification: Requirements gathering and analysis, Software Requirements Specification (SRS), Formal system specification, Axiomatic specification, Algebraic specification, Executable specification and 4GL.

UNIT- III:

Software Design: Overview of the design process, How to characterize a good software design? Layered arrangement of modules, Cohesion and Coupling. approaches to software design.

Agility: Agility and the Cost of Change, Agile Process, Extreme Programming (XP), Other Agile Process Models, Tool Set for the Agile Process (Text Book 2)

Function-Oriented Software Design: Overview of SA/SD methodology, Structured analysis, Developing the DFD model of a system, Structured design, Detailed design, and Design Review.

User Interface Design: Characteristics of a good user interface, Basic concepts, Types of user interfaces, Fundamentals of component-based GUI development, and user interface design methodology.

UNIT- IV:Coding And Testing: Coding, Code review, Software documentation, Testing, Black-box testing, White-Box testing, Debugging, Program analysis tools, Integration testing**,** Testing object-oriented programs, Smoke testing, and Some general issues associated with testing.

Software Reliability And Quality Management: Software reliability. Statistical testing, Software quality, Software quality management system, ISO 9000. SEI Capability maturity model. Few other important quality standards, and Six Sigma**.**

UNIT-V: Computer-Aided Software Engineering (Case): CASE and its scope, CASE environment, CASE support in the software life cycle, other characteristics of CASE tools, Towards second generation CASE Tool, and Architecture of a CASE Environment.
Software Maintenance: Characteristics of software maintenance, Software reverse engineering, Software maintenance process models and Estimation of maintenance cost.
Software Reuse: reuse- definition, introduction, reason behind no reuse so far, Basic issues in any reuse program, A reuse approach, and Reuse at organization level.

Text Books:

1. Fundamentals of Software Engineering, Radii b Mall, 5th Edition, PHI.
2. Software Engineering A practitioner's Approach, Roger S. Pressman, 9th Edition, Mc- Grow Hill International Edition.

Reference Books:

1. Software Engineering, Ian Sommerville,10th Edition, Pearson.
2. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.

e- Resources:

1) https://nptel.ac.in/courses/106/105/106105182/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01260589506387148827_shared/overview
3) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013382690411003904735_shared/overview

III B.Tech I Semester

23ECT16	**WIRELESS SENSOR NETWORKS FOR CS (Professional Elective-1)**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Course Objectives:

- To acquire the knowledge about various architectures and applications of Sensor Networks
- To understand issues, challenges and emerging technologies for wireless sensor networks
- To learn about various routing protocols and MAC Protocols
- To understand various data gathering and data dissemination methods
- To Study about design principals, node architectures, hardware and software required for implementation of wireless sensor networks.

Course Outcomes: Upon completion of the course, the student will be able to:

- Analyze and compare various architectures of Wireless Sensor Networks
- Understand Design issues and challenges in wireless sensor networks
- Analyze and compare various data gathering and data dissemination methods.
- Design, Simulate and Compare the performance of various routing and MAC protocol

UNIT - I:

Introduction to Sensor Networks, unique constraints and challenges, Advantage of Sensor Networks, Applications of Sensor Networks, Types of wireless sensor networks

UNIT - II:

Mobile Ad-hoc Networks (MANETs) and Wireless Sensor Networks, Enabling technologies for Wireless Sensor Networks. Issues and challenges in wireless sensor networks

UNIT - III:

Routing protocols, MAC protocols: Classification of MAC Protocols, S-MAC Protocol, B-MAC protocol, IEEE 802.15.4 standard and Zombie

UNIT - IV:

Dissemination protocol for large sensor network. Data dissemination, data gathering, and data fusion; Quality of a sensor network; Real-time traffic support and security protocols.

UNIT - V:

Design Principles for WSNs, Gateway Concepts Need for gateway, WSN to Internet Communication, and Internet to WSN Communication. Single-node architecture, Hardware components & design constraints, Operating systems and execution environments, introduction to Tinos and neck.

TEXT BOOKS:

1. Ad-Hoc Wireless Sensor Networks- C. Siva Ram Murthy, B. S. Manoj, Pearson
2. Principles of Wireless Networks – Kavas Palaver and P. Krishna Murthy, 2002, PE

REFERENCE BOOKS:

1. Wireless Digital Communications – Kaila Fehr, 1999, PHI.
2. Wireless Communications-Andrea Goldsmith, 2005 Cambridge University Press.
3. Mobile Cellular Communication – Gestapo Sasibhushana Rao, Pearson Education, 2012.
4. Wireless Communication and Networking – William Stallings, 2003, PHI.

III B.Tech I Semester

23ECT12	**MICROPROCESSORS AND MICROCONTROLLERS**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Course Objectives:

1. To learn the fundamental architectural concepts of microprocessors.
2. To gain knowledge about assembly language programming concepts.
3. To get familiar about 8086 interfacing.
4. To understand the fundamentals of the 8051 Microcontroller.
5. To learn interfacing with the 8051 Microcontroller.

Course Outcomes:

At the end of this course, the students will be able to

1. Learn the fundamental architectural concepts of microprocessors.
2. Gain knowledge about assembly language programming concepts.
3. Understand the concepts of 8086 interfacing.
4. Learn the fundamentals of the 8051 Microcontroller.
5. Know the interfacing with the 8051 Microcontroller.

UNIT I

8086 Architecture: Main features, pin diagram/description, 8086 microprocessor family, internal architecture, bus interfacing unit, execution unit, interrupts and interrupt response, 8086 system timing, minimum mode and maximum mode configuration.

UNIT II

8086 Programming: Program development steps, instructions, addressing modes, assembler directives, writing simple programs with an assembler, assembly language program development tools.

UNIT III

8086 Interfacing: Semiconductor memories interfacing (RAM, ROM), Intel 8255 programmable peripheral interface, Interfacing switches and LEDS, Interfacing seven segment displays, software and hardware interrupt applications, Intel 8251 USART architecture and interfacing, Intel 8237a DMA controller, stepper motor, A/D and D/A converters, Need for 8259 programmable interrupt controllers.

UNIT IV

Microcontroller : Architecture of 8051 – Special Function Registers(SFRs) - I/O Pins Ports and Circuits - Instruction set - Addressing modes - Assembly language programming.

UNIT V

Interfacing Microcontroller :- Programming 8051 Timers - Serial Port Programming - Interrupts Programming – LCD & Keyboard Interfacing - ADC, DAC & Sensor Interfacing - External Memory Interface- Stepper Motor and Waveform generation - Comparison of Microprocessor, Microcontroller, PIC and ARM processors

Textbooks:

1. Microprocessors and Interfacing – Programming and Hardware by Douglas V Hall, SSSP Rao, Tata McGraw Hill Education Private Limited, 3rdEdition,1994.
2. K M Bhurchandi, A K Ray, Advanced Microprocessors and Peripherals, 3rd edition, McGraw Hill Education, 2017.
3. Raj Kamal, Microcontrollers: Architecture, Programming, Interfacing and System Design, 2nd edition, Pearson, 2012.

References:

1. Ramesh S Gaonkar, Microprocessor Architecture Programming and Applications with the 8085, 6th edition, Penram International Publishing, 2013.
2. Kenneth J. Ayala, The 8051 Microcontroller, 3rd edition, Cengage Learning, 2004.

III B.Tech I Semester

		L	T	P	C
23CST08	**ARTIFICIAL INTELLIGENCE (Professional Elective-1)**	**3**	**0**	**0**	**3**

Pre-requisite:

- Knowledge in Computer Programming.
- A course on ―Mathematical Foundations of Computer Science‖.
- Background in linear algebra, data structures and algorithms, and probability.

Course Objectives:

- The student should be made to study the concepts of Artificial Intelligence.
- The student should be made to learn the methods of solving problems using Artificial Intelligence.
- The student should be made to introduce the concepts of Expert Systems.
- To understand the applications of AI, namely game playing, theorem proving, and machine learning.
- To learn different knowledge representation techniques

UNIT - I

Introduction: AI problems, foundation of AI and history of AI intelligent agents: Agents and Environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.

UNIT - II

Searching- Searching for solutions, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Hill climbing, A* ,AO* Algorithms, Problem reduction, Game Playing-Adversial search, Games, mini-max algorithm, optimal decisions in multiplayer games, Problem in Game playing, Alpha-Beta pruning, Evaluation functions.

UNIT - III

Representation of Knowledge: Knowledge representation issues, predicate logic- logic programming, semantic nets- frames and inheritance, constraint propagation, representing knowledge using rules, rules based deduction systems. Reasoning under uncertainty, review of probability, Bayes‘ probabilistic interferences and dempstershafer theory.

UNIT - IV

Logic concepts: First order logic. Inference in first order logic, propositional vs. first order inference, unification & lifts forward chaining, Backward chaining, Resolution, Learning from observation Inductive learning, Decision trees, Explanation based learning, Statistical Learning methods, Reinforcement Learning.

UNIT - V

Expert Systems: Architecture of expert systems, Roles of expert systems – Knowledge Acquisition Meta knowledge Heuristics. Typical expert systems – MYCIN, DART, XCON: Expert systems shells.

Textbooks:

1. S. Russel and P. Norvig, ―Artificial Intelligence – A Modern Approach‖, SecondEdition, Pearson Education.
2. Kevin Night and Elaine Rich, Nair B., ―Artificial Intelligence (SIE)‖, Mc Graw Hill

Reference Books:

1. David Poole, Alan Mack worth, Randy Goebel,‖ Computational Intelligence: a logical approach‖, Oxford University Press.
2. G. Luger, ―Artificial Intelligence: Structures and Strategies for complex problemsolving‖, Fourth Edition, Pearson Education.
3. J. Nilsson, ―Artificial Intelligence: A new Synthesis‖, Elsevier Publishers.
4. Artificial Intelligence, SarojKaushik, CENGAGE Learning.

Online Learning Resources:

1. https://ai.google/
2. https://swayam.gov.in/nd1_noc19_me71/preview

III B.Tech I Semester

23CSP16	**CLOUD COMPUTING LAB**	**L**	**T**	**P**	**C**
		0	**0**	**3**	**1.5**

Course Objectives:

- Demonstrate application development using Cloud
- Explain features of Hardtop

Course Outcomes (CO):

On completion of this course, the students will be able to:

- Configure various virtualization tools such as Virtual Box, VMware workstation.
- Design and deploy a web application in a Peas environment.
- Learn how to simulate a cloud environment to implement new schedulers.
- Install and use a generic cloud environment that can be used as a private cloud.
- Manipulate large data sets in a parallel environment.

List of Experiments:

1. Install VirtualBox/VMware Workstation with different flavours of Linux or windows OS on top of windows operating systems.
2. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs
3. Install Google App Engine. Create hello world app and other simple web applications using python/java.
4. Use GAE launcher to launch the web applications.
5. Simulate a cloud scenario using Clouds and run a scheduling algorithm that is not present in Cloud Shim.
6. Find a procedure to transfer the files from one virtual machine to another virtual machine.
7. Find a procedure to launch virtual machine using try stack (Online Open stack Demo Version)
8. Install Hardtop single node cluster and run simple applications like word count
9. Establish an AWS account. Use the AWS Management Console to launch an EC2 instance and connect to it.
10. Develop a Guestbook Application using Google App Engine
11. Develop a Server less Web App using AWS
12. Design a Content Recommendation system using AWS
13. Design a Cloud based smart traffic management system
14. Design Cloud based attendance management system
15. Design E-learning cloud-based system
16. Using Amazon Lax build a Chabot

References:

1. https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html
2. http://code.google.com/appengine/downloads.html
3. http://code.google.com/appengine/downloads.html

Online Learning Resources/Virtual Labs:

1. Google Cloud Computing Foundations Course - Course (nptel.ac.in)

III B.Tech I Semester

23CYP02	**CYBER SECURITY LAB**	**L**	**T**	**P**	**C**
		0	**0**	**3**	**3**

Course Objectives:

To get practical exposure of Cyber security threats and Forensics tools.

Course Outcomes:

1. Get the skill to identify cyber threats/attacks.
2. Get the knowledge to solve security issues in day-to-day life.
3. Able to use Autopsy tools
4. Perform Memory capture and analysis
5. Demonstrate Network analysis using Network miner tools

List of Experiments

1. Perform an Experiment for port scanning with neap
2. Set Up a honey pot and monitor the honey pot on the network
3. Install Jscript/Cryptal tool (or any other equivalent) and demonstrate Asymmetric, symmetric crypto algorithm, Hash and Digital/PKI signatures.
4. Generate minimum 10 passwords of length 12 characters using open SSL command
5. Perform practical approach to implement Foot printing-Gathering target information using Dmitry-Dmagic, U Attester
6. Working with sniffers for monitoring network communication (Wire shark).
7. Using Snort, perform real time traffic analysis and packet logging.
8. Perform email analysis using the Autopsy tool.
9. Perform Registry analysis and get boot time logging using process monitor tool
10. Perform File type detection using Autopsy tool
11. Perform Memory capture and analysis using FTK imager tool
12. Perform Network analysis using the Network Miner tool

TEXT BOOKS:

1. Real Digital Forensics for Handheld Devices, E. P. Dorothy, Auer back Publications, 2013.
2. The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics, J. Sammons, Singles Publishing, 2012.

REFERENCE BOOKS:

1. Handbook of Digital Forensics and Investigation, E. Casey, Academic Press, 2010.
2. Malware Forensics Field Guide for Windows Systems: Digital Forensics Field Guides, C. H. Malian, E. Casey and J. M. Aquiline, Signers, 2012.
3. The Best Damn Cybercrime and Digital Forensics Book.

III B.Tech I Semester

23CSP12	**FULL STACK DEVELOPMENT – II (Skill Enhancement Course)**	**L**	**T**	**P**	**C**
		0	**1**	**2**	**2**

Course Objectives: The main objectives of the course are to

- Make use of Modern- day JavaScript with ES6 standards for designing Dynamic web pages
- Building robust & responsive User Interfaces using popular JavaScript library **‗React.js'**. Building robust backend APIs using **‗Express. js'**
- Establishing the connection between frontend (React) User interfaces and backend APIs (Express) with Data Bases(My SQL)
- Familiarize students with GitHub for remote repository hosting and collaborative development.

Course Outcomes:

- CO1: Building fast and interactive UIs
- CO2: Applying Declarative approach for developing web apps
- CO3: Understanding ES6 features to embrace modern JavaScript
- CO4: Building reliable APIs with Express. Js
- CO5: Create and manage Git repositories, track changes, and push code to GitHub.

Experiments covering the Topics:

- Introduction to DOM (Document Object Model), Ecma Script (ES6) standards and features like Arrow functions, Spread operator, Rest operator, Type coercion, Type hoisting, String literals, Array and Object Destructuring.
- Basics of React. js like React Components, JSX, Conditional rendering
  Differences between Real DOM and Virtual DOM.
- Important React.js concepts like React hooks, Props, React forms, Fetch API, Iterative rendering using JavaScript map() function.
- JavaScript runtime environment node. js and its uses, Express. js and Routing, Micro-Services architecture and MVC architecture, database connectivity using (My SQL)
- Introduction to My SQL, setting up MySQL and configuring, Databases, My SQL queries, subqueries, creating My SQL driver for database connectivity to Express. js server.
- Introduction to Git and GitHub and upload project& team collaboration

Sample Experiments:

1. **Introduction to Modern JavaScript and DOM**
   - **a.** Write a JavaScript program to link JavaScript file with the HTML page
   - **b.** Write a JavaScript program to select the elements in HTML page using selectors
   - **c.** Write a JavaScript program to implement the event listeners
   - **d.** Write a JavaScript program to handle the click events for the HTML button elements
   - **e.** Write a JavaScript program to with three types of functions
     - **i.** Function declaration
     - **ii.** Function definition

iii. Arrow functions

2. Basics of React. js
   - **a.** Write a React program to implement a counter button using react class components
   - **b.** Write a React program to implement a counter button using react functional components
   - **c.** Write a React program to handle the button click events in functional component
   - **d.** Write a React program to conditionally render a component in the browser
   - **e.** Write a React program to display text using String literals
3. Important concepts of React. js
   - **a.** Write a React program to implement a counter button using React use State hook
   - **b.** Write a React program to fetch the data from an API using React use Effect hook
   - **c.** Write a React program with two react components sharing data using Props.
   - **d.** Write a React program to implement the forms in react
   - **e.** Write a React program to implement the iterative rendering using map() function.

4. Introduction to Git and GitHub

a. **Setup**
   - Install Git on local machine.
   - Configure Git (user name, email).
   - Create GitHub account and generate a personal access token.

b. **Basic Git Workflow**
   - Create a local repository using git init
   - Create and add files → git add .
   - Commit files → git commit -m "Initial commit"
   - Connect to GitHub remote → git remote add origin <repo_url>
   - Push to GitHub → git push -u origin main

c. **Branching and Collaboration**
   - Create a branch → git checkout -b feature1
   - Merge branch to main → git merge feature1
   - Resolve merge conflicts (guided)

5. Upload React Project to GitHub

   - Create a new React app using npx create-react-app myapp
   - Initialize a git repo and push to GitHub
   - Use .gitignore to exclude node_modules
   - Create multiple branches: feature/navbar, feature/form
   - Practice merge and pull requests (can use GitHub GUI)

6. Introduction to Node. js and Express. js
   - **a.** Write a program to implement the _hello world‘ message in the route through the browser using Express
   - **b.** Write a program to develop a small website with multiple routes using Express. js
   - **c.** Write a program to print the _hello world‘ in the browser console using Express. js
   - **d.** Write a program to implement the CRUD operations using Express. js
   - **e.** Write a program to establish the connection between API and Database using Express – My SQL driver
7. Introduction to My SQL

a. Write a program to create a Database and table inside that database using My SQL Command line client
b. Write a My SQL queries to create table, and insert the data, update the data in the table
c. Write a My SQL queries to implement the subqueries in the My SQL command line client
d. Write a My SQL program to create the script files in the My SQL workbench
e. Write a My SQL program to create a database directory in Project and initialize a database. sql file to integrate the database into API

8. Team Collaboration Using GitHub

- Form groups of 2–3 students
- Create a shared GitHub repo
- Assign tasks and work in branches
- Use Issues, Pull Requests, and Code Reviews
- Document code with README.md

Textbooks:

1. Web Design with HTML, CSS, JavaScript and JQuery Set Book by Jon Duckett Professional JavaScript for Web Developers Book by Nicholas C. Zakas
2. John Dean, Web Programming with HTML5, CSS and JavaScript, Jones & Bartlett Learning, 2019.
3. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O'Reilly.
4. Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites by Robin Nixon
5. AZAT MARDAN, Full Stack Java Script: Learn Back bone. js, Node.jsand Mongo DB.2015

Reference Books:

1. Full-Stack JavaScript Development by Eric Bush.
2. Programming the World Wide Web, 7th Edition, Robet W Sebesta, Pearson, 2013.
3. Tomasz Dyl ,KamilPrzeorski , MaciejCzarnecki, Mastering Full Stack React Web Development 2017

Online Learning Resources:

1. https://ict.iitk.ac.in/product/full-stack-developer-html5-css3-js-bootstrap-php-4/
2. https://www.w3schools.com/html
3. https://www.w3schools.com/css
4. https://www.w3schools.com/js/
5. https://www.w3schools.com/nodejs
6. https://www.w3schools.com/typescript
7. https://docs.github.com/
8. https://education.github.com/git-cheat-sheet-education.pdf

III B.Tech – I semester

L	T	P	C
0	0	2	1

(23ECP09) - TINKERING LAB

The aim of tinkering lab for engineering students is to provide a hands-on learning environment where students can explore, experiment, and innovate by building and testing prototypes. These labs are designed to demonstrate practical skills that complement theoretical knowledge.

	Course objectives: The objectives of the course are to
1	Encourage Innovation and Creativity
2	Provide Hands-on Learning and Impart Skill Development
3	Foster Collaboration and Teamwork
4	Enable Interdisciplinary Learning, Prepare for Industry and Entrepreneurship
5	Impart Problem-Solving mind-set

These labs bridge the gap between academia and industry, providing students with the practical experience. Some students may also develop entrepreneurial skills, potentially leading to start-ups or innovation-driven careers. Tinkering labs aim to cultivate the next generation of engineers by giving them the tools, space, and mind-set to experiment, innovate, and solve real-world challenges.

List of experiments:

1) Make your own parallel and series circuits using breadboard for any application of your choice.
2) Design and 3D print a Walking Robot
3) Design and 3D Print a Rocket.
4) Temperature & Humidity Monitoring System (DHT11 + LCD)
5) Water Level Detection and Alert System
6) Automatic Plant Watering System
7) Bluetooth-Based Door Lock System
8) Smart Dustbin Using Ultrasonic Sensor
9) Fire Detection and Alarm System
10) RFID-Based Attendance System
11) Voice-Controlled Devices via Google Assistant
12) Heart Rate Monitoring Using Pulse Sensor
13) Soil Moisture-Based Irrigation
14) Smart Helmet for Accident Detection
15) Milk Adulteration Detection System
16) Water Purification via Activated Carbon
17) Solar Dehydrator for Food Drying
18) Temperature-Controlled Chemical Reactor
19) Ethanol Mini-Plant Using Biomass
20) Smart Fluid Flow Control (Solenoid + pH Sensor)
21) Portable Water Quality Tester
22) AI Crop Disease Detection
23) AI-based Smart Irrigation
24) ECG Signal Acquisition and Plotting
25) AI-Powered Traffic Flow Prediction

26) Smart Grid Simulation with Load Monitoring
27) Smart Campus Indoor Navigator
28) Weather Station Prototype
29) Firefighting Robot with Sensor Guidance
30) Facial Recognition Dustbin
31) Barcode-Based Lab Inventory System
32) Growth Chamber for Plants
33) Biomedical Waste Alert System
34) Soil Classification with AI
35) Smart Railway Gate
36) Smart Bin Locator via GPS and Load Sensors
37) Algae-Based Water Purifier
38) Contactless Attendance via Face Recognition

- **Note:** The students can also design and implement their own ideas, apart from the list of experiments mentioned above.
- **Note:** A minimum of 8 to 10 experiments must be completed by the students.

III B.Tech II Semester (CSE-Cyber Security)

S.No.	Course Code	Title	L	T	P	Credits
1	23CYT03	Cyber Crimes & Digital Forensics	3	0	0	3
2	23CST18	Cryptography & Network Security	3	0	0	3
3	23CST17	Machine Learning	3	0	0	3
4	23CST19a 23CST19c 23CST23d 23CYT04	**Professional Elective-II** 1. Software Testing Methodologies 2. DevOps 3. Internet of Things 4. Applied Cryptography	3	0	0	3
5	23CST20a 23CST20b 23CST20c 23CYT05	**Professional Elective-III** 1. Software Project Management 2. Mobile Adhoc Networks 3. Natural Language Processing 4. Security Assessment and Risk Analysis	3	0	0	3
6		**Open Elective – II**	3	0	0	3
7	23CYP03	Cryptography & Network Security Lab	0	0	3	1.5
8	23CYP04	Cyber Crimes & Digital Forensics Lab	0	0	3	1.5
9	23BSP07	**Skill Enhancement course-IV** Soft skills	0	1	2	2
10	23BST28	**Audit Course** Technical Paper Writing & IPR	2	0	0	-
		Total	**20**	**1**	**08**	**23**
Mandatory Industry Internship of 08 weeks duration during summer vacation						

12 week MOOC Swayam/NPTEL course recommended by the BoS

S.No.	Course Name	Link
1	Internet of Things	**https://onlinecourses.nptel.ac.in/noc25_cs44/preview**
2	Applied Cryptography	https://onlinecourses.nptel.ac.in/noc25_cs31/
3	Natural Language Processing	https://onlinecourses.nptel.ac.in/noc25_cs51/

Open Elective – II

S.No.	Course Code	Course Name	Offered by the Dept.
1	23CET19	Disaster Management	CIVIL
2	23CET20	Sustainable in Engineering Practices	
3	23EET18	Renewable Energy Sources	EEE
4	23MET19	Automation and Robotics	ME
5	23ECT25	Digital Electronics	ECE
6	23BST24	Operations Research	Mathematics
7	23BST25	Physics Of Electronic Materials and Devices	Physics
8	23BST26	Chemistry Of Polymers and Applications	Chemistry
9	23BST27	Academic Writing and Public Speaking	Humanities
10	23BST28	Technical paper writing and Intellectual proper rights	Humanities
11	23BST29	Mathematical foundation of Quantum Technologies	Mathematics

III B.Tech II Semester

23CYT03	Cyber Crimes & Digital Forensics	L	T	P	C
		3	0	0	3

Course Objectives:
1. To analyze how to conduct a digital forensics investigation and validate forensics data.
2. Understand the impact of cyber crime in real time applications
3. Discover the various methods to find the cyber crime
4. Apply the forensics technology
5. Develop the laws to control cyber crime

Course Outcomes:
1. Understand the fundamentals of cybercrime.
2. Understand the various cybercrime issues.
3. Understand different investigation tools for cybercrime.
4. Understand basics of Forensic Technology and Practices.
5. Analyze different laws, ethics and evidence handling procedures.

UNIT-I: Introduction **9 Hrs**
Introduction and Overview of Cyber Crime, Nature and Scope of Cyber Crime, Types of Cyber Crime: Social Engineering, Categories of Cyber Crime, Property Cyber Crime.

UNIT-II: Cyber Crime Issues **9 Hrs**
Unauthorized Access to Computers, Computer Intrusions, White collar Crimes, Viruses and Malicious Code, Internet Hacking and Cracking, Virus Attacks, Pornography, Software Piracy, Intellectual Property, Mail Bombs, Exploitation, Stalking and Obscenity in Internet, Digital laws and legislation, Law Enforcement Roles and Responses.

UNIT-III: Investigation **9 Hrs**
Introduction to Cyber Crime Investigation, Investigation Tools, rediscovery, Digital Evidence Collection, Evidence Preservation, E-Mail Investigation, E-Mail Tracking, IP Tracking, E- Mail Recovery, Hands on Case Studies. Encryption and Decryption Methods, Search and Seizure of Computers, Recovering Deleted Evidences, Password Cracking.

UNIT-IV: Digital Forensics **9Hrs**
Introduction to Digital Forensics, Forensic Software and Hardware, Analysis and Advanced Tools, Forensic Technology and Practices, Forensic Ballistics and Photography, Face, Iris and Fingerprint Recognition, Audio Video Analysis, Windows System Forensics, Linux System Forensics, Network Forensics.

UNIT-V: Laws and Acts **9 Hrs**
Laws and Ethics, Digital Evidence Controls, Evidence Handling Procedures, Basics of Indian Evidence ACT IPC and Corps, Electronic Communication Privacy ACT, Legal Policies

TEXT BOOKS:
1. Nelson Phillips and Engineer Stuart, ―Computer Forensics and Investigations‖, CEng age Learning, New Delhi, 2009.
2. Kevin Media, Chris Promise, Matt Pipe, ―Incident Response and Computer Forensics ―, Tata McGraw -Hill, New Delhi, 2006.

REFERENCE BOOKS:
1. Robert M Slade,‖ Software Forensics‖, Tata McGraw - Hill, New Delhi, 2005.
2. Bernadette H Schell, Clemens Martin, ―Cybercrime‖, ABC – CLIO Inc, California, 2004
3. ―Understanding Forensics in IT ―, NIIT Ltd, 2005.

III B.Tech II Semester

23CYT04	APPLIED CRYPTOGRAPHY (Professional Elective -II)	L	T	P	C
		3	0	0	3

Course Objectives: Knowledge on significance of cryptographic protocols and symmetric and public key algorithms

Course Outcomes:

1. Understand the various cryptographic protocols
2. Analyze key length and algorithm types and modes
3. Illustrate different public key algorithms in cryptosystems
4. Understand special algorithms for protocols and usage in the real world.

UNIT - I

Foundations: Terminology, Steganography, Substitution Ciphers and Transposition Ciphers, Simple XOR, One-Time Pads, Computer Algorithms, Large Numbers,

Cryptographic Protocols: Protocol Building Blocks: Introduction to Protocols, Communications Using Symmetric Cryptography, One-Way Functions, One-Way Hash Functions, Communications Using Public-Key Cryptography, Digital Signatures, Digital Signatures with Encryption, Random and Pseudo-Random-Sequence Generation

UNIT - II

Cryptographic Techniques: Key length: Symmetric Key length, Public key length, comparing symmetric and public key length.

Algorithm types and modes: Electronic Codebook Mode, Block Replay, Cipher Block Chaining Mode, Stream Cipher, Self-Synchronizing Stream Ciphers, Cipher-Feedback Mode, Synchronous Stream Ciphers, Output-Feedback Mod, Counter Mode, Other Block-Cipher Modes.

UNIT - III

Public-Key Algorithms: Background, Knapsack Algorithms, RSA, Poling-Hellman, Rabin, El Gamma, Mc Ellice, Elliptic Curve Cryptosystems, LUC, Finite Automaton Public-Key Cryptosystems

Public-Key Digital Signature Algorithms: Digital Signature Algorithm (DSA), DSA Variants, Ghost Digital Signature Algorithm, Discrete Logarithm Signature Schemes, Gong - Schnorr-Shamir, ESIGN

UNIT - IV

Special Algorithms for Protocols: Multiple-Key Public-Key Cryptography, Secret-Sharing Algorithms, Subliminal Channel, Undeniable Digital Signatures, Designated Confirmer Signatures, Computing with Encrypted Data, Fair Coin Flips, One-Way Accumulators, All-or-Nothing Disclosure of Secrets, Fair and Failsafe Cryptosystems, Zero-Knowledge Proofs of Knowledge, Blind Signatures, Oblivious Transfer,

Secure Multiparty Computation, Probabilistic Encryption, Quantum Cryptography

UNIT - V

Real World Approaches: IBM Secret key management protocol, ISDN, Kerberos, Krypto Knight, Privacy enhanced mail (PEM), Message security protocol (MSP), PGP, Public-Key Cryptography Standards (PKCS), Universal Electronic Payment System (UEPS).

TEXT BOOKS:

1. Bruce Schneider, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C (cloth)

III B.Tech II Semester

23CYT05	Security Assessment and Risk Analysis (Professional Elective-III)	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES

1. The course takes a software development perspective to the challenges of engineering software systems that are secure.
2. This course addresses design and implementation issues critical to producing secure software systems.
3. The course deals with the question of how to make the requirements for confidentiality, integrity, and availability integral to the software development process.
4. Secure software requirements gathering to design, development, configuration, deployment, and ongoing maintenance
5. Security of enterprise information systems.

COURSE OUTCOMES:

1. Understand various aspects and principles of software security.
2. Devise security models for implementing at the design level.
3. Identify and analyze the risks associated with s/w engineering and use relevant models to mitigate the risks.
4. Understand the various security algorithms to implement for secured computing and computer networks
5. Explain different security frameworks for different types of systems including electronic systems.

UNIT-I: Introduction **9 Hrs**

Defining computer security, the principles of secure software, trusted computing base, etc, threat modelling, advanced techniques for mapping security requirements into design specifications. Secure software implementation, deployment and ongoing management.

UNIT-II: Security Design **9 Hrs**

Software design and an introduction to hierarchical design representations. Difference between high-level and detailed design. Handling security with high- level design. General Design Notions. Security concerns designs at multiple levels of abstraction, Design patterns, quality assurance activities and strategies that support early vulnerability detection, Trust models, security Architecture & design reviews.

UNIT-III: Software Assurance Model **9 Hrs**

Identify project security risks & selecting risk management strategies, Risk Management Framework, Security Best practices/ Known Security Flaws, Architectural risk analysis, Security Testing & Reliability (Penn testing, Risk- Based Security Testing

UNIT-IV: Application **9 Hrs**

Software Security in Enterprise Business: Identification and authentication, Enterprise Information Security, Symmetric and asymmetric cryptography, including public key cryptography, data encryption standard (DES), advanced encryption standard (AES), algorithms for hashes and message digests. Authentication, authentication schemes , access control models, Kerberos protocol, public key infrastructure (PKI), protocols specially designed for e-commerce and web applications, firewalls and VPNs

UNIT-V: Security Frameworks **9 Hrs**

Security development frameworks. Security issues associated with the development and deployment of information systems, including Internet-based e-commerce, e-business, and e- service systems.

TEXT BOOKS:

1. C. Kaufman, R. Perlman, & M. Spicier, Network security: Private communication in a public world, 2nd Edition, Upper Saddle River, Prentice Hall, 2002

2. Charles P. P fleeter , Shari Lawrence Fleeter, Security in Computing, 4th Edition, Upper Saddle River, NJ: Prentice Hall, 2007

3. W. Stallings, Cryptography and network security: Principles and practice, 5th Edition, Upper Saddle River, NJ: Prentice Hall., 2011

4. M. Marko, & J. Breat aupt, Information security: Principles and practices. Upper Saddle River, NJ: Prentice Hall, 2005

REFERENCE BOOKS:

1. Gary McGraw, Software Security: Building Security In, Addison-Wesley, 2006

III B.Tech II Semester

23CYP03	CRYPTOGRAPHY AND NETWORK SECURITY LAB	L	T	P	C
		0	0	3	1.5

List of Experiments:

1. Write a C program that contains a string (char pointer) with a value ‗Hello world'. The program should XOR each character in this string with 0 and displays the result.
2. Write a C program that contains a string (char pointer) with a value ‗Hello world'. The program should AND or and XOR each character in this string with 127 and display the result.
3. Write a Java program to perform encryption and decryption using the following algorithms
   a. Cease cipher b. Substitution cipher c. Hill Cipher
4. Write a C/JAVA program to implement the DES algorithm logic.
5. Write a C/JAVA program to implement the Blowfish algorithm logic.
6. Write a C/JAVA program to implement the Randal algorithm logic.
7. Write the RC4 logic in Java Using Java cryptography; encrypt the text ―Hello world‖ using Blowfish. Create your own key using Java key tool.
8. Write a Java program to implement RSA algorithm.
9. Implement the Daffy-Hellman Key Exchange mechanism using HTML and JavaScript.
10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
11. Calculate the message digest of a text using the MD5 algorithm in JAVA.

III B.Tech II Semester

23CYP04	Cyber Crimes & Digital Forensics Lab	L	T	P	C
		0	0	3	1.5

Course Description: This course is designed to give students the tools and techniques for investigating crime involving digital evidence.

Course Objectives:

1. To provide students with a comprehensive overview of collecting, investigating, preserving, and presenting evidence of cybercrime left in digital storage devices, emails, browsers, mobile devices using different Forensics tools.
2. To Understand file system basics and where hidden files may lie on the disk, as well as how to extract the data and preserve it for analysis.
3. Understand some of the tools of e-discovery.
4. To understand the network analysis, Registry analysis
5. Analyze attacks using different forensics tools.

Course Outcomes:

1. Learn the importance of a systematic procedure for investigation of data found on digital storage media that might provide evidence of wrong-doing.
2. To Learn the file system storage mechanisms and retrieve files in hidden format.
3. Learn the use of computer forensics tools used in data analysis.
4. Learn how to find data that may be clear or hidden on a computer disk, find out the open ports for the attackers through network analysis, Registry analysis.
5. Apply different computer forensic tools to a given cybercrime scene

List of Experiments

1. Perform email analysis using the tools like Exchange EDB viewer, MBOX viewer and View user mailboxes and public folders, Filter the mailbox data based on various criteria, Search for particular items in user mailboxes and public folders
2. Perform Browser history analysis and get the downloaded content, history, saved logins, searches, websites visited etc. using Foxton Forensics tool, Dumpzilla.
3. Perform mobile analysis in the form of retrieving call logs, SMS log, all contacts list using the forensics tool like SAFT
4. Perform Registry analysis and get boot time logging using process monitor tool
5. Perform Disk imaging and cloning the using the X-way Forensics tools
6. Perform Data Analysis i.e History about open file and folder, and view folder actions using Last view activity tool
7. Perform Network analysis using the Network Miner tool.
8. Perform information for incident response using the crowd Response tool
9. Perform File type detection using Autopsy tool
10. Perform Memory capture and analysis using the Live RAM capture or any forensic tool

TEXT BOOKS:

1. Real Digital Forensics for Handheld Devices, E. P. Dorothy, Auer back Publications, 2013.
2. The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics, J. Sammons, Singers Publishing, 2012