



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

R23 B.Tech Cyber Security

(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)

III Year I Semester	CLOUD COMPUTING	L	T	P	C
		3	0	0	3

#### Course Objectives:

The main objectives of the course are to

- Explain the evolving utility computing model called cloud computing.
- Introduce the various levels of services offered by cloud.
- Discuss the fundamentals of cloud enabling technologies such as distributed computing, service-oriented architecture and virtualization.
- Emphasize the security and other challenges in cloud computing.
- Introduce the advanced concepts such as containers, serverless computing and cloud-centric Internet of Things.

#### UNIT -I: Introduction to Cloud Computing Fundamentals

Cloud computing at a glance, defining a cloud, cloud computing reference model, types of services (IaaS, PaaS, SaaS), cloud deployment models (public, private, hybrid), utility computing, cloud computing characteristics and benefits, cloud service providers (Amazon Web Services, Microsoft Azure, Google AppEngine).

#### UNIT-II: Cloud Enabling Technologies

Ubiquitous Internet, parallel and distributed computing, elements of parallel computing, hardware architectures (SISD, SIMD, MISD, MIMD), elements of distributed computing, technologies for distributed computing, IPC, RPC, Messaging systems, Service Oriented Architecture (SOA), Web services, virtualization.

#### UNIT-III: Virtualization and Containers

Characteristics of virtualized environments, taxonomy of virtualization techniques, virtualization and cloud Computing, pros and cons of virtualization, technology examples (XEN, VMware), building blocks of containers, container platforms (LXC, Docker), container orchestration, Docker Swarm and Kubernetes, public cloud VM (e.g. Amazon EC2) and container (e.g. Amazon Elastic Container Service) offerings.

#### UNIT-IV: Cloud computing challenges

Economics of the cloud, cloud interoperability and standards, scalability and fault tolerance, energy efficiency in clouds, federated clouds, cloud computing security, fundamentals of computer security, cloud security architecture, cloud shared responsibility model, security in cloud deployment models.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

**UNIT -V:** Advanced concepts in cloud computing

Serverless computing, Function-as-a-Service, serverless computing architecture, public cloud (e.g. AWS Lambda) and open-source (e.g. OpenFaaS), serverless platforms, Internet of Things (IoT), applications, cloud-centric IoT and layers, edge and fog computing, DevOps, infrastructure-as-code, quantum cloud computing.

**Text Books:**

1. Mastering Cloud Computing, 2<sup>nd</sup> edition, Rajkumar Buyya, Christian Vecchiola, ThamaraiSelvi, ShivanandaPoojara, Satish N. Srirama, Mc Graw Hill, 2024.
2. Distributed and Cloud Computing, Kai Hwang, Geoffery C. Fox, Jack J. Dongarra, Elsevier, 2012.

**Reference Books:**

1. Cloud Computing, Theory and Practice, Dan C Marinescu, 2<sup>nd</sup> edition, MK Elsevier, 2018.
2. Essentials of cloud Computing, K. Chandrasekhran, CRC press, 2014.
3. Online documentation and tutorials from cloud service providers (e.g., AWS, Azure, GCP)



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

<b>III Year I Semester</b>	<b>CRYPTOGRAPHY &amp; NETWORK SECURITY</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### **Course Objectives:**

The main objectives of this course are to

- Explore the working principles and utilities of various cryptographic algorithms including secret key cryptography, hashes and message digests, public key algorithms, design issues and working principles of various authentication protocols.
- Implement various secure communication standards including Kerberos, IPsec, and SSL/TLS.

### **UNIT I:**

**Basic Principles:** Security Goals, Cryptographic Attacks, Services and Mechanisms, Mathematics of Cryptography- integer arithmetic, modular arithmetic, matrices, linear congruence.

### **UNIT II:**

**Symmetric Encryption:** Mathematics of Symmetric Key Cryptography-algebraic structures, GF( $2^n$ ) Fields, Introduction to Modern Symmetric Key Ciphers-modern block ciphers, modern stream ciphers, Data Encryption Standard- DES structure, DES analysis, Security of DES, Multiple DES, Advanced Encryption Standard-transformations, key expansions, AES ciphers, Analysis of AES.

### **UNIT III:**

**Asymmetric Encryption:** Mathematics of Asymmetric Key Cryptography-primes, primality testing, factorization, CRT, Asymmetric Key Cryptography- RSA crypto system, Rabin cryptosystem, Elgamal Crypto system, ECC

### **UNIT IV:**

**Data Integrity, Digital Signature Schemes & Key Management:** Message Integrity and Message Authentication-message integrity, Random Oracle model, Message authentication, Cryptographic Hash Functions-whirlpool, SHA-512, Digital Signature-process, services, attacks, schemes, applications, Key Management-symmetric key distribution, Kerberos.

### **UNIT V:**

**Network Security-I:** Security at application layer: PGP and S/MIME, Security at the Transport Layer: SSL and TLS, **Network Security-II :**Security at the Network Layer: IPsec-two modes, two security protocols, security association, IKE, ISAKMP, System Security-users, trust, trusted systems, buffer overflow, malicious software, worms, viruses, IDS, Firewalls.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

**Text Books:**

1. Cryptography and Network Security, 3<sup>rd</sup> Edition Behrouz A Forouzan, Deb deep Mukhopadhyay, McGraw Hill,2015
2. Cryptography and Network Security,4<sup>th</sup> Edition, William Stallings, (6e) Pearson,2006
3. Everyday Cryptography, 1<sup>st</sup> Edition, Keith M.Martin, Oxford,2016

**Reference Books:**

Network Security and Cryptography, 1<sup>st</sup> Edition, Bernard Meneges, Cengage Learning,2018



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

R23 B.Tech Cyber Security

(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)

<b>III Year I Semester</b>	<b>AUTOMATA THEORY &amp; COMPILER DESIGN</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### Course Objectives:

- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- To understand deterministic and non-deterministic machines and the differences between decidability and undecidability.
- Introduce the major concepts of language translation and compiler design and impart the knowledge of practical skills necessary for constructing a compiler.
- Topics include phases of compiler, parsing, syntax directed translation, type checking use of symbol tables, intermediate code generation.

### Course Outcomes:

- Able to employ finite state machines for modeling and solving computing problems.
- Able to design context free grammars for formal languages.
- Able to distinguish between decidability and undecidability.
- Demonstrate the knowledge of patterns, tokens & regular expressions for lexical analysis.
- Acquires skills in using lex tool and design LR parsers

### UNIT-I

**Introduction to Finite Automata:** Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory—Alphabets, Strings, Languages, Problems. **Non deterministic Finite Automata:** Formal Definition, an application, Text Search, Finite Automata with Epsilon-Transitions. **Deterministic Finite Automata:** Definition of DFA, How DFA Process Strings, The language of DFA, Conversion of NFA with  $\epsilon$ -transitions to NFA without  $\epsilon$ -transitions. Conversion of NFA to DFA

### UNIT-II

**Regular Expressions:** Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions. **Pumping Lemma for Regular Languages:** Statement of the pumping lemma, Applications of the Pumping Lemma. **Context-Free Grammars:** Definition of Context-Free Grammars, Derivations Using a Grammar, Left most and Right most Derivations, the Language of a Grammar, Parse Trees, Ambiguity in Grammars and Languages.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

### **UNIT-III**

**Push Down Automata:** Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state. **Turing Machines:** Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine. **Undecidability:** Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines

### **UNIT-IV**

**Introduction:** The structure of a compiler, **Lexical Analysis:** The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical-Analyzer Generator Lex. **Syntax Analysis:** Introduction, Context-Free Grammars, Writing a Grammar, Top-Down Parsing, Bottom-Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers.

### **UNIT-V**

**Syntax-Directed Translation:** Syntax-Directed Definitions, Evaluation Orders for SDD's, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's. **Intermediate-Code Generation:** Variants of Syntax Trees, Three-Address Code. **Run Time Environments:** Stack Allocation of Space, Access to Non local Data on the Stack, Heap Management

### **TEXT BOOKS:**

1. Introduction to Automata Theory, Languages, and Computation, 3<sup>nd</sup> Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman, Pearson Education.
2. Compilers: Principles, Techniques and Tools, Alfred V.Aho, Monica S.Lam, RaviSethi, Jeffry D.Ullman, 2<sup>nd</sup> Edition, Pearson.
3. Theory of Computer Science— Automata languages and computation, MishraandChandrashekaran, 2<sup>nd</sup> Edition, PHI.

### **REFERENCE BOOKS:**

1. Introduction to Formal languages Automata Theory and Computation, Kamala Krithivasan, Rama R, Pearson.
2. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
3. Lex &yacc—John R. Levine, Tony Mason, Doug Brown, O'reilly
4. Compiler Construction, Kenneth C. Louden, Thomson. Course Technology.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

<b>III Year I Semester</b>	<b>SECURE CODING PRACTICES</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### **COURSE OBJECTIVES:**

1. To understand of the various security attacks and knowledge to recognize and remove common coding errors that lead to vulnerabilities.
2. It gives an outline of the techniques for developing a secure application.

### **COURSE OUTCOMES:**

Upon the completion of the course, the students will be able to:

- CO 1:** Implement security as a culture and show mistakes that make applications vulnerable to attacks.
- CO 2:** Understand various attacks like DoS, buffer overflow, web specific, database specific web-spoofing attacks.
- CO 3:** Analyze and demonstrate skills necessary to address common programming errors that lead to security issues and to learn how to develop secure applications.
- CO 4:** Analyze the nature of the threats to software and incorporate secure coding practices throughout the planning and development of the product.
- CO 5:** Apply proper techniques for handling application faults, implement secure authentication, authorization and data validation controls used to prevent common vulnerabilities.

### **UNIT-I**

**INTRODUCTION: Need for secure systems:** Proactive Security development process, Secure Software Development Cycle (S-SDLC), Security issues while writing SRS, Design phase security, Development Phase, Test Phase, Maintenance Phase, Writing Secure Code – Best Practices SD3 (Secure by design, default and deployment), Security principles and Secure Product Development Timeline

### **UNIT-II**

**SECURE CODING TECHNIQUES:** Protection against DoS attacks, Application Failure Attacks, CPU Starvation Attacks, Insecure Coding Practices in Java Technology. ARP Spoofing and its countermeasures. Buffer Overrun- Stack overrun, Heap Overrun, Array Indexing Errors, Format String Bugs. Security Issues in C Language: String Handling, Avoiding Integer Overflows and Underflows and Type Conversion Issues- Memory Management Issues, Code Injection Attacks, Canary based countermeasures using StackGuard and Propolice. Socket Security, Avoiding Server Hijacking, Securing RPC, ActiveX and DCOM



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

**UNIT-III**

**Threat modelling process and its benefits:** Identifying the Threats by Using Attack Trees and rating threats using DREAD, Risk Mitigation Techniques and Security Best Practices. Security techniques, authentication, authorization. Defense in Depth and Principle of Least Privilege.

**UNIT-IV**

**AND WEB SPECIFIC INPUT ISSUES:** SQL Injection Techniques and Remedies, Race conditions, Time of Check Versus Time of Use and its protection mechanisms. Validating Input and Inter process Communication, Securing Signal Handlers and File Operations. XSS scripting attack and its types – Persistent and Non persistent attack XSS Countermeasures and Bypassing the XSS Filters. Testing Secure Applications: Security code overview, secure software installation. The Role of the Security Tester, Building the Security Test Plan. Testing HTTP-Based Applications, Testing File-Based Applications, Testing Clients with Rogue Servers

**UNIT -V**

**SOFTWARE SECURITY ENGINEERING:** Requirements engineering for secure software: Misuse and abuse cases- SQUARE process model- Software security practices and knowledge for architecture and design.

**Text Books:**

1. Writing Secure Code, Michael Howard and David LeBlanc, Microsoft Press, 2nd Edition, 2004.
2. Threat Modeling, Frank Swiderski and Window Snyder, Microsoft Professional, 1<sup>st</sup> Edition, 2004.

**Reference Books:**

1. Robert C.Seacord, “ *Secure Coding in C and C++*”, Pearson Education, 2nd edition, 2013.
2. Julia H. Allen, Sean J. Barnum, Robert J. Ellison, Gary McGraw, Nancy R. Mead, “ *Software Security Engineering : A guide for Project Managers*”, Addison-Wesley Professional, 2008.
3. Buffer Overflow Attacks: Detect, Exploit, Prevent by Jason Deckar, Syngress, 1<sup>st</sup> Edition, 2005.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

<b>III Year I Semester</b>	<b>SOFTWARE ENGINEERING</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### **Objectives:**

The objectives of this course are to introduce

- Software life cycle models
- Software requirements and SRS document.
- How to plan for a project.
- The quality control and how to ensure good quality software.
- Testing Course methods of software, use of CASE tools
- Implementation issues, validation and verification procedures.

### **Course Outcomes:**

<b>CO</b>	<b>Course Outcomes</b>	<b>Knowledge Level(K) #</b>
<b>CO1</b>	Compare and analyse various process models	<b>K1</b>
<b>CO2</b>	Develop SRS document and estimate the modularity of the project	<b>K2</b>
<b>CO3</b>	Develop data flow diagrams and compare the user interface design	<b>K2</b>
<b>CO4</b>	Compare testing strategies and analyse the software quality	<b>K3</b>
<b>CO5</b>	Apply Computer Aided Software Engineering tools and analyse the components of software maintenance and reuse.	<b>K2</b>

#basedonsuggestedRevisedBTL

### **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 and Spiral model.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

#### **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, and 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 characterise a good software design? Layered arrangement of modules, Cohesion and Coupling. approaches to software design. **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:** What can be reused? Why almost no reuse so far? Basic issues in any reuse program, A reuse approach, and Reuse at organisation level.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

### **Text Books**

1. Fundamentals of Software Engineering, Raji b Mall, Fifth Edition, PHI.

### **Reference Books**

1. Software Engineering A practitioner's Approach, Roger S. Pressman, Ninth Edition, McGraw Hill International Edition.
2. Software Engineering, Ian Sommerville, Tenth Edition, Pearson Education.
3. 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](https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01260589506387148827_shared/overview)  
[https://infyspringboard.onwingspan.com/web/en/app/toc/lex\\_auth\\_013382690411003904735\\_shared/overview](https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013382690411003904735_shared/overview)



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

<b>III Year I Semester</b>	<b>ARTIFICIAL INTELLIGENCE</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### **Course Outcome(s):**

This course introduces students to the basic knowledge representation, problem solving, and learning methods of artificial intelligence.

### **UNIT-I**

**Introduction, Overview of Artificial intelligence:** Problems of AI, AI technique, Tic - Tac - Toe problem. Intelligent Agents, Agents & environment, nature of environment, structure of agents, goal-based agents, utility-based agents, learning agents. **Problem Solving, Problem Space & search:** Defining the problem as state space search, production system, problem characteristics, issues in the design of search programs.

### **UNIT-II**

**Search techniques:** Problem solving agents, searching for solutions; uniform search strategies: breadth first search, depth first search, depth limited search, bidirectional search, comparing uniform search strategies. Heuristic search strategies Greedy best-first search, A\* search, AO\* search, memory bounded heuristic search: local search algorithms & optimization problems: Hill climbing search, simulated annealing search, local beam search

### **UNIT-III**

**Constraint satisfaction problems:** Local search for constraint satisfaction problems. Adversarial search, Games, optimal decisions & strategies in games, the minimax search procedure, alpha-beta pruning, additional refinements, iterative deepening.

### **UNIT – IV**

**Knowledge & reasoning:** Knowledge representation issues, representation & mapping, approaches to knowledge representation. Using predicate logic, representing simple fact in logic, representing instant & ISA relationship, computable functions & predicates, resolution, natural deduction. Representing knowledge using rules, Procedural verses declarative knowledge, logic programming, forward verses backward reasoning, matching, control knowledge.

### **UNIT – V**

**Probabilistic reasoning:** Representing knowledge in an uncertain domain, the semantics of Bayesian networks, Dempster-Shafer theory, Planning Overview, components of a planning system, Goal stack planning, Hierarchical planning, other planning techniques. **Expert Systems:** Representing and using domain knowledge, expert system shells, and knowledge acquisition.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

**Home Assignments:**

Assignments should include problems related to the topics covered in lectures, like heuristics, optimal search, and graph heuristics. Constraint satisfaction problems, k-nearest neighbors, decision trees, etc. can be included in home assignments.

**Text Books:**

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach
2. Artificial Intelligence, Russel, Pearson

**Reference Books:**

1. Artificial Intelligence, Ritch & Knight, TMH
2. Introduction to Artificial Intelligence & Expert Systems, Patterson, PHI
3. Logic & Prolog Programming, Saroj Kaushik, New Age International
4. Expert Systems, Giarranto, VIKAS



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

<b>III Year I Semester</b>	<b>MICROPROCESSORS &amp; MICROCONTROLLERS</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		3	0	0	3

**Course Objectives:**

- To introduce fundamental architectural concepts of microprocessors and microcontrollers.
- To impart knowledge on addressing modes and instruction set of 8086 and 8051
- To introduce assembly language programming concepts
- To explain memory and I/O interfacing with 8086 and 8051
- To introduce 16 bit and 32 bit microcontrollers.

**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, 3<sup>rd</sup> Edition, 1994.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

2. K M Bhurchandi, A K Ray, Advanced Microprocessors and Peripherals, 3<sup>rd</sup> edition, McGraw Hill Education, 2017.
3. Raj Kamal, Microcontrollers: Architecture, Programming, Interfacing and System Design, 2<sup>nd</sup> edition, Pearson, 2012.

**References:**

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



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

<b>III Year I Semester</b>	<b>CLOUD COMPUTING LAB</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>

**Course Objectives:**

- To introduce the various levels of services offered by cloud.
- To give practical knowledge about working with virtualization and containers.
- To introduce the advanced concepts such as serverless computing and cloud simulation.

**Course Outcomes:** At the end of the course, the student should be able to

- Demonstrate various service types, delivery models and technologies of a cloud computing environment.
- Distinguish the services based on virtual machines and containers in the cloud offerings.
- Assess the challenges associated with a cloud-based application.
- Discuss advanced cloud concepts such as serverless computing and cloud simulation.
- Examine various programming paradigms suitable to solve real world and scientific problems using cloud services.

**List of Experiments:**

1. Lab on web services
2. Lab on IPC, messaging, publish/subscribe
3. Install VirtualBox/VMware Workstation with different flavours of Linux or windows OS on top of windows8 or above.
4. Install a C compiler in the virtual machine created using VirtualBox and execute Simple Programs.
5. Create an Amazon EC2 instance and set up a web-server on the instance and associate an IP address with the instance. In the process, create a security group allowing access to port 80 on the instance.

OR

6. Do the same with OpenStack
7. Install Google App Engine. Create a hello world app and other simple web applications using python/java.
8. Start a Docker container and set up a web-server (e.g. apache2 or Python based Flask micro web framework) on the instance. Map the host directory as a data volume for the container.
9. Find a procedure to transfer the files from one virtual machine to another virtual machine. Similarly, from one container to another container.
10. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
11. Install Hadoop single node cluster and run simple applications like word count.
12. Utilize OpenFaaS – Serverless computing framework and demonstrate basic event driven function invocation.
13. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

**Text Books:**

1. Mastering Cloud Computing, 2<sup>nd</sup> edition, RajkumarBuyya, Christian Vecchiola, ThamaraiSelvi, ShivanandaPoojara, Satish N. Srirama, McGraw Hill, 2024.
2. Distributed and Cloud Computing, Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra, Elsevier, 2012.

**Reference Books:**

1. Cloud Computing, Theory and Practice, Dan C Marinescu, 2<sup>nd</sup> edition, MK Elsevier, 2018.
2. Cloud Computing: Principles and Paradigms by RajkumarBuyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
3. Online documentation and tutorials from cloud service providers (e.g. AWS, Google App Engine)
4. Docker, Reference documentation, <https://docs.docker.com/reference/>  
OpenFaaS, Serverless Functions Made Simple, <https://docs.openfaas.com/>



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA–533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

<b>III Year II Semester</b>	<b>CRYPTOGRAPHY &amp; NETWORK SECURITY LAB</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		0	0	3	1.5

**Course Objectives:**

- To learn basic understanding of cryptography, how it has evolved, and some key encryption techniques used today.
- To understand and implement encryption and decryption using Ceaser Cipher, Substitution Cipher, Hill Cipher.

**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) Ceaser Cipher
  - b) Substitution Cipher
  - c) Hill Cipher
4. Write a 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 Rijndael algorithm logic.
7. 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 Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).
10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA–533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

III Year I Semester	FULLSTACK DEVELOPMENT - II	L	T	P	C
		0	1	2	2

**Course Outcomes:**

At the end of the Course, Student will be able to:

- CO1: Make use of router, template engine and authentication using sessions to develop application in ExpressJS.
- CO2: Build a single page application using RESTful APIs in ExpressJS.
- CO3: Make use of components, props, stats and render data in ReactJS.
- CO4: Apply router and hooks in designing ReactJS application.
- CO5: Make use of MongoDB queries to perform CRUD operations on document database.

**Mapping of course outcomes with program outcomes:**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	2	2	3		2				2	2		2
CO 2	2	3	2		2				2	2		2
CO 3	2	2	3		2				2	2		2
CO 4	2	3	2		2				2	2		2
CO 5	2	2	3		2				2	2		2

**List of Experiments:**

**Experiment 1:Node.js**

- a. Write a program to show the workflow of JavaScript code executable by creating web server in Node.js.
- b. Write a program to transfer data over http protocol using http module.
- c. Create a text file src.txt and add the following content to it. (HTML, CSS, Javascript, Typescript, MongoDB, Express.js, React.js, Node.js)
- d. Write a program to parse an URL using URL module.
- e. Write a program to create an user-defined module and show the workflow of Modularization of application using Node.js

**Experiment 2:TypeScript**

- a. Write a program to understand simple and special types.
- b. Write a program to understand function parameter and return types.
- c. Write a program to show the importance with Arrow function. Use optional, default and REST parameters.
- d. Write a program to understand the working of typescript with class, constructor, properties, methods and access specifiers.
- e. Write a program to understand the working of namespaces and modules.
- f. Write a program to understand generics with variables, functions and constraints.

**Experiment 3-15:**

Augmented Programs: (Any 2 must be completed from Experiment 3-5)

- 3. Write a CSS program, to apply 2D and 3D transformations in a web page.
- 4. a web page with new features of HTML5 and CSS3.
- 5. Design a to-do list application using JavaScript.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA–533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

**Experiment 6:**

**ExpressJS – Routing, HTTP Methods, Middleware**

- a. Write a program to define a route, Handling Routes, Route Parameters, Query Parameters and URL building.
- b. Write a program to accept data, retrieve data and delete a specified resource using http methods.
- c. Write a program to show the working of middleware.

**Experiment 7:**

**ExpressJS – Templating, Form Data**

- a. Write a program using templating engine.
- b. Write a program to work with form data.

**Experiment 8:**

**ExpressJS – Cookies, Sessions, Authentication**

- a. Write a program for session management using cookies and sessions.
- b. Write a program for user authentication

**Experiment 9:**

**ExpressJS – Database, RESTful APIs**

- a. Write a program to connect MongoDB database using Mongoose and perform CRUD operations.
- b. Write a program to develop a single page application using RESTful APIs

**Experiment 10:**

**ReactJS – Render HTML, JSX, Components – function & Class**

- a. Write a program to render HTML to a web page.
- b. Write a program for writing markup with JSX.
- c. Write a program for creating and nesting components (function and class).

**Experiment 11:**

**ReactJS – Props and States, Styles, Respond to Events**

- a. Write a program to work with props and states.
- b. Write a program to add styles (CSS & Sass Styling) and display data.
- c. Write a program for responding to events.

**Experiment 12:**

**ReactJS – Conditional Rendering, Rendering Lists, React Forms**

- a. Write a program for conditional rendering.
- b. Write a program for rendering lists.
- c. Write a program for working with different form fields using react forms



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA–533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

**Experiment 13:**

**ReactJS – React Router, Updating the Screen**

- a. Write a program for routing to different pages using react router.
- b. Write a program for updating the screen.

**Experiment 14:**

**ReactJS – Hooks, Sharing data between Components**

- a. Write a program to understand the importance of using hooks.
- b. Write a program for sharing data between components

**Experiment 15:**

**ReactJS Applications – To-do list and Quiz**

- a. Design to-do list application

**Experiment 16:**

**MongoDB – Installation, Configuration, CRUD operations**

- a. Install MongoDB and configure ATLAS
- b. Write MongoDB queries to perform CRUD operations on document using insert(), find(), update(), remove()

**Experiment 17:**

**MongoDB – Databases, Collections and Records**

- g. Write MongoDB queries to Create and drop databases and collections.

Write MongoDB queries to work with records using find(), limit(), sort(), createIndex(), aggregate()

**Experiment 18-20:**

Augmented Programs: (Any 2 must be completed)

18. Design a to-do list application using NodeJS and ExpressJS.
19. Design a Quiz app using ReactJS.
20. Complete the MongoDB certification from MongoDB University website.

**Text Books:**

1. Programming the World Wide Web, 7th Edition, Robet W Sebesta, Pearson, 2013.

2. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2<sup>nd</sup> edition, A Press, O'Reilly.

**Web Links:**

1. ExpressJS - <https://www.tutorialspoint.com/expressjs>

2. ReactJS - <https://www.w3schools.com/REACT> (and) <https://react.dev/learn#>

3. MongoDB - <https://learn.mongodb.com/learning-paths/introduction-to-mongodb>



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

KAKINADA-533003, Andhra Pradesh, India

**R23 B.Tech Cyber Security**

**(R23 – III<sup>rd</sup> YEAR COURSE STRUCTURE & SYLLABUS)**

III Year I Semester	UI DESIGN - FLUTTER LAB	L	T	P	C
		0	0	2	1

**Course Objectives:**

- Learns to Implement Flutter Widgets and Layouts
- Understands Responsive UI Design and with Navigation in Flutter
- Knowledge on Widgets and customize widgets for specific UI elements, Themes
- Understand to include animation apart from fetching data

**List of Experiments:**

Students need to implement the following experiments

1. a) Install Flutter and Dart SDK.  
b) Write a simple Dart program to understand the language basics.
2. a) Explore various Flutter widgets (Text, Image, Container, etc.).  
b) Implement different layout structures using Row, Column, and Stack widgets.
3. a) Design a responsive UI that adapts to different screen sizes.  
b) Implement media queries and breakpoints for responsiveness.
4. a) Set up navigation between different screens using Navigator.  
b) Implement navigation with named routes.
5. a) Learn about stateful and stateless widgets.  
b) Implement state management using set State and Provider.
6. a) Create custom widgets for specific UI elements.  
b) Apply styling using themes and custom styles.
7. a) Design a form with various input fields.  
b) Implement form validation and error handling.
8. a) Add animations to UI elements using Flutter's animation framework.  
b) Experiment with different types of animations (fade, slide, etc.).
9. a) Fetch data from a REST API.  
b) Display the fetched data in a meaningful way in the UI.
10. a) Write unit tests for UI components.  
b) Use Flutter's debugging tools to identify and fix issues.

**Text Book:**

1. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development.
2. Rap Payne, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps 1<sup>st</sup> Edition, Apres