C201 DISCRETE MATHEMATICS

C201.1: Have knowledge of the concepts needed to test the logic of a program.

C201.2: Have an understanding in identifying structures on many levels.

C201.3: Be aware of a class of functions which transform a finite set into another finite set which relates to input and output functions in computer science.

C201.4: Be aware of the counting principles.

C201.5: Be exposed to concepts and properties of algebraic structures such as groups, rings and fields.

C202 DIGITAL PRINCIPLES AND SYSTEM DESIGN

C202.1: Simplify Boolean functions using KMap

C202.2: Design and Analyze Combinational and Sequential Circuits

C202.3: Implement designs using Programmable Logic Devices

C202.4: Write HDL code for combinational and Sequential Circuits

C203 DATA STRUCTURES

C203.1: Implement abstract data types for linear data structures.

C203.2: Apply the different linear and non-linear data structures to problem solutions.

C203.3: Critically analyze the various sorting algorithms

C204 OBJECT ORIENTED PROGRAMMING

C204.1: Develop Java programs using OOP principles

C204.2: Develop Java programs with the concepts inheritance and interfaces

C204.3: Build Java applications using exceptions and I/O streams

C204.4: Develop Java applications with threads and generics classes

C204.5: Develop interactive Java programs using swings

C205 COMMUNICATION ENGINEERING

C205.1: Ability to comprehend and appreciate the significance and role of this course in the present contemporary world

C205.2: Apply analog and digital communication techniques.

C205.3: Use data and pulse communication techniques.

C205.4: Analyze Source and Error control coding

C206 DATA STRUCTURES LABORATORY

C206.1: Write functions to implement linear and non-linear data structure operations

C206.2: Suggest appropriate linear / non-linear data structure operations for solving a given problem

C206.3: Appropriately use the linear / non-linear data structure operations for a given problem

C206.4: Apply appropriate hash functions that result in a collision free scenario for data storage and retrieval

C207 OBJECT ORIENTED PROGRAMMING LABORATORY

C207.1: Develop and implement Java programs for simple applications that make use of classes, packages and interfaces.

C207.2: Develop and implement Java programs with arraylist, exception handling and multithreading

C207.3: Design applications using file processing, generic programming and event handling

C208 DIGITAL SYSTEMS LABORATORY

C208.1: Implement simplified combinational circuits using basic logic gates

C208.2: Implement combinational circuits using MSI devices

C208.3: Implement sequential circuits like registers and counters

C208.4: Simulate combinational and sequential circuits using HDL

C209 INTERPERSONAL SKILLS/LISTENING&SPEAKING

C209.1: Listen and respond appropriately.

C209.2: Participate in group discussions

C209.3: Make effective presentations

C209.4: Participate confidently and appropriately in conversations both formal and informal

C210 PROBABILITY AND QUEUING THEORY

C210.1: Understand the fundamental knowledge of the concepts of probability and have knowledge of standard distributions which can describe real life phenomenon.

C210.2: Understand the basic concepts of one and two dimensional random variables and apply in engineering applications.

C210.3: Apply the concept of random processes in engineering disciplines.

C210.4: Acquire skills in analyzing queueing models.

C210.5: Understand and characterize phenomenon which evolve with respect to time in a probabilistic manner

C211 COMPUTER ARCHITECTURE

C211.1: Understand the basics structure of computers, operations and instructions.

C211.2: Design arithmetic and logic unit.

C211.3: Understand pipelined execution and design control unit.

C211.4: Understand parallel processing architectures.

C211.5: Understand the various memory systems and I/O communication

C212 DATABASE MANAGEMENT SYSTEMS

C212.1: Classify the modern and futuristic database applications based on size and complexity

C212.2: Map ER model to Relational model to perform database design effectively

C212.3: Write queries using normalization criteria and optimize queries

C212.4: Compare and contrast various indexing strategies in different database systems

C212.5: Appraise how advanced databases differ from traditional databases

C213 DESIGN AND ANALYSIS OF ALGORITHMS

C213.1: Design algorithms for various computing problems.

C213.2: Analyze the time and space complexity of algorithms.

C213.3: Critically analyze the different algorithm design techniques for a given problem.

C213.4: Modify existing algorithms to improve efficiency

C214 OPERATING SYSTEMS

C214.1: Analyze various scheduling algorithms.

C214.2: Understand deadlock, prevention and avoidance algorithms.

C214.3: Compare and contrast various memory management schemes.

C214.4: Understand the functionality of file systems.

C214.5: Perform administrative tasks on Linux Servers.

C214.6: Compare iOS and Android Operating Systems

C215 SOFTWARE ENGINEERING

C215.1: Identify the key activities in managing a software project.

C215.2: Compare different process models.

C215.3: Concepts of requirements engineering and Analysis Modeling.

C215.4: Apply systematic procedure for software design and deployment.

C215.5: Compare and contrast the various testing and maintenance.

C215.6: Manage project schedule, estimate project cost and effort required

C216 DATABASE MANAGEMENT SYSTEMS LABORATORY

C216.1: Use typical data definitions and manipulation commands.

C216.2: Design applications to test Nested and Join Queries

C216.3: Implement simple applications that use Views

C216.4: Implement applications that require a Front-end Tool

C216.5: Critically analyze the use of Tables, Views, Functions and Procedures

C217 OPERATING SYSTEMS LABORATORY

C217.1: Compare the performance of various CPU Scheduling Algorithms

C217.2: Implement Deadlock avoidance and Detection Algorithms

C217.3: Implement Semaphores

C217.4: Create processes and implement IPC

C217.5: Analyze the performance of the various Page Replacement Algorithms

C217.6 Implement File Organization and File Allocation Strategies

C218 ADVANCED READING AND WRITING

C218.1: Write different types of essays.

C218.2: Write winning job applications.

C218.3: Read and evaluate texts critically.

C218.4: Display critical thinking in various professional contexts

C301 ALGEBRA AND NUMBER THEORY

C301.1: Apply the basic notions of groups, rings, fields which will then be used to solve related problems.

C301.2: Explain the fundamental concepts of advanced algebra and their role in modern mathematics and applied contexts.

C301.3: Demonstrate accurate and efficient use of advanced algebraic techniques.

C301.4: Demonstrate their mastery by solving non - trivial problems related to the concepts, and by proving simple theorems about the, statements proven by the text.

C301.5: Apply integrated approach to number theory and abstract algebra, and provide a firm basis for further reading and study in the subject

C302 COMPUTER NETWORKS

C302.1: Understand the basic layers and its functions in computer networks.

C302.2: Evaluate the performance of a network.

C302.3: Understand the basics of how data flows from one node to another.

C302.4: Analyze and design routing algorithms.

C302.5: Design protocols for various functions in the network.

C302.6: Understand the working of various application layer protocols

C303 MICROPROCESSORS AND MICROCONTROLLERS

C303.1: Understand and execute programs based on 8086 microprocessor.

C303.2: Design Memory Interfacing circuits.

C303.3: Design and interface I/O circuits.

C303.4: Design and implement 8051 microcontroller based systems

C304 THEORY OF COMPUTATION

C304.1: Construct automata, regular expression for any pattern.

C304.2: Write Context free grammar for any construct.

C304.3: Design Turing machines for any language.

C304.4: Propose computation solutions using Turing machines.

C304.5: Derive whether a problem is decidable or not

C305 OBJECT ORIENTED ANALYSIS AND DESIGN

C305.1: Express software design with UML diagrams

C305.2: Design software applications using OO concepts.

C305.3: Identify various scenarios based on software requirements

C305.4: Transform UML based software design into pattern based design using design patterns

C305.5: Understand the various testing methodologies for OO software

C306 MICROPROCESSORS AND MICROCONTROLLERS LABORATORY

C306.1: Write ALP Programmes for fixed and Floating Point and Arithmetic operations

C306.2: Interface different I/Os with processor

C306.3: Generate waveforms using Microprocessors

C306.4: Execute Programs in 8051

C306.5: Explain the difference between simulator and Emulator

C307 OBJECT ORIENTED ANALYSIS AND DESIGN LABORATORY

C307.1: Perform OO analysis and design for a given problem specification.

C307.2: Identify and map basic software requirements in UML mapping.

C307.3: Improve the software quality using design patterns and to explain the rationale behind applying specific design patterns

C307.4: Test the compliance of the software with the SRS

C308 NETWORKS LABORATORY

C308.1: Implement various protocols using TCP and UDP.

C308.2: Compare the performance of different transport layer protocols.

C308.3: Use simulation tools to analyze the performance of various network protocols.

C308.4: Analyze various routing algorithms.

C308.5: Implement error correction codes.

C309 INTERNET PROGRAMMING

C309.1: Construct a basic website using HTML and Cascading Style Sheets.

C309.2: Build dynamic web page with validation using Java Script objects and by applying different event handling mechanisms.

C309.3: Develop server side programs using Servlets and JSP.

C309.4: Construct simple web pages in PHP and to represent data in XML format.

C309.5: Use AJAX and web services to develop interactive web applications

C310 ARTIFICIAL INTELLIGENCE

C310.1: Use appropriate search algorithms for any AI problem

C310.2: Represent a problem using first order and predicate logic

C310.3: Provide the apt agent strategy to solve a given problem

C310.4: Design software agents to solve a problem

C310.5: Design applications for NLP that use Artificial Intelligence

C311 MOBILE COMPUTING

C311.1: Explain the basics of mobile telecommunication systems

C311.2: Illustrate the generations of telecommunication systems in wireless networks

C311.3: Determine the functionality of MAC, network layer and Identify a routing protocol for a given Ad hoc network

C311.4: Explain the functionality of Transport and Application layers

C311.5: Develop a mobile application using android/blackberry/ios/Windows SDK

C312 COMPILER DESIGN

C312.1: Understand the different phases of compiler.

C312.2: Design a lexical analyzer for a sample language.

C312.3: Apply different parsing algorithms to develop the parsers for a given grammar.

C312.4: Understand syntax-directed translation and run-time environment.

C312.5: Learn to implement code optimization techniques and a simple code generator.

C312.6: Design and implement a scanner and a parser using LEX and YACC tools

C313 DISTRIBUTED SYSTEMS

C313.1: Elucidate the foundations and issues of distributed systems

C313.2: Understand the various synchronization issues and global state for distributed systems.

C313.3: Understand the Mutual Exclusion and Deadlock detection algorithms in distributed systems

C313.4: Describe the agreement protocols and fault tolerance mechanisms in distributed systems.

C313.5: Describe the features of peer-to-peer and distributed shared memory systems

C314 INTERNET PROGRAMMING LABORATORY

C314.1: Construct Web pages using HTML/XML and style sheets.

C314.2: Build dynamic web pages with validation using Java Script objects and by applying different event handling mechanisms.

C314.3: Develop dynamic web pages using server side scripting.

C314.4: Use PHP programming to develop web applications.

C314.5: Construct web applications using AJAX and web services

C315 MOBILE APPLICATION DEVELOPMENT LABORATORY

C315.1: Develop mobile applications using GUI and Layouts.

C315.2: Develop mobile applications using Event Listener.

C315.3: Develop mobile applications using Databases.

C315.4: Develop mobile applications using RSS Feed, Internal/External Storage, SMS, Multi-threading and GPS.

C315.5: Analyze and discover own mobile app for simple needs

C401 PRINCIPLES OF MANAGEMENT

C401.1: Upon completion of the course, students will be able to have clear understanding of managerial functions like planning, organizing, staffing, leading & controlling and have same basic knowledge on international aspect of management

C402 CRYPTOGRAPHY AND NETWORK SECURITY

C402.1: Understand the fundamentals of networks security, security architecture, threats and vulnerabilities

C402.2: Apply the different cryptographic operations of symmetric cryptographic algorithms

C402.3: Apply the different cryptographic operations of public key cryptography

C402.4: Apply the various Authentication schemes to simulate different applications.

C402.5: Understand various Security practices and System security standards

C403 CLOUD COMPUTING

C403.1: Articulate the main concepts, key technologies, strengths and limitations of cloud computing.

C403.2: Learn the key and enabling technologies that help in the development of cloud.

C403.3: Develop the ability to understand and use the architecture of compute and storage cloud, service and delivery models.

C403.4: Explain the core issues of cloud computing such as resource management and security.

C403.5: Be able to install and use current cloud technologies.

C403.6: Evaluate and choose the appropriate technologies, algorithms and approaches for implementation and use of cloud

C404 CLOUD COMPUTING LABORATORY

C404.1: Configure various virtualization tools such as Virtual Box, VMware workstation.

C404.2: Design and deploy a web application in a PaaS environment.

C404.3: Learn how to simulate a cloud environment to implement new schedulers.

C404.4: Install and use a generic cloud environment that can be used as a private cloud.

C404.5: Manipulate large data sets in a parallel environment

C405 SECURITY LABORATORY

C405.1: Develop code for classical Encryption Techniques to solve the problems.

C405.2: Build cryptosystems by applying symmetric and public key encryption algorithms.

C405.3: Construct code for authentication algorithms.

C405.4: Develop a signature scheme using Digital signature standard.

C405.5: Demonstrate the network security system using open source tools

C406 – Project Work

C406.1: Students will acquire the ability to make links across different areas of knowledge and to generate, develop and evaluate ideas and information so as to apply these skills to the project tasks

C406.2: Students will acquire the skills to critically analyze the problem, communicate effectively and to present ideas clearly and coherently to specific audience in both the written and forms.

C406.3: Students will acquire collaborative skills through working in a team to achieve common goals.

C406.4: Students will be able to learn on their own, reflect on their learning and take appropriate actions to improve it