KAMARAJ COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution - AFFILIATED TO ANNA UNIVERSITY, CHENNAI)

S.P.G.Chidambara Nadar - C.Nagammal Campus

S.P.G.C. Nagar, K.Vellakulam – 625 701 (Near VIRUDHUNAGAR).

B.TECH. INFORMATION TECHNOLOGY

REGULATIONS – 2021

AUTONOMOUS SYLLABUS

CHOICE BASED CREDIT SYSTEM

CURRICULUM AND SYLLABI

(Semester III and IV)

Vision of the Department

To make the department of Information Technology the unique of its kind in the field of Research and Development activities in this part of world

Mission of the Department

To impart highly innovative and technical knowledge in the field of Information Technology to the urban and unreachable rural student folks through Total Quality Education.

Program Educational Objectives (PEOs)

PEO1: Technical Knowledge: Graduates will be able to identify, analyze and create solutions for real life, industrial and societal needs by applying the principles and practices of Information Technology.

PEO2: Teamwork & Ethics: Graduates will be able to collaborate effectively and ethically in a multi-disciplinary team as a member &/ as a leader.

PEO3: Lifelong Learning: Graduates will be able to adopt the contemporary technologies in the field of Information Technology to provide solutions for challenging environments.

Program Specific Objectives (PSOs)

PSO 1: Demonstrate technical and interpersonal skills to design and develop IT enabled solutions to meet the real time industrial and societal needs

PSO2: Exhibit an ability to adapt to the evolutionary changes in computing

SEMESTER III

SL NO	COURSE CODE	COURSE TITLE	CATEGORY	TOTAL CONTACT PERIODS	L	T	P	C
THEORY								
1	MA2201	Linear Algebra and Boundary Value Problems	BS	4	3	1	0	4
2	IT2201	Computer Organization and Architecture	PC	3	3	0	0	3
3	IT2202	Object Oriented Programming	PC	3	3	0	0	3
4	IT2203	Software Engineering	PC	3	3	0	0	3
5	EC2203	Digital Systems	ES	3	3	0	0	3
6	EE2201	Fundamentals of Electrical and Electronics Engineering	ES	3	3	0	0	3
7	GE2201	Design Thinking	ES	3	3	0	0	3
8		Audit Course	AUD	3	3	0	0	0
PRACTICALS								
9	IT2204	Object Oriented Programming Laboratory	PC	4	0	0	4	2
10	EC2204	Digital Systems Laboratory	ES	4	0	0	4	2
			TOTAL	**33**	**24**	**1**	**8**	**26**

SEMESTER IV

SL NO	COURSE CODE	COURSE TITLE	CATEGORY	TOTAL CONTACT PERIODS	L	T	P	C
THEORY								
1	MA2251	Discrete Mathematics & Probability	BS	4	3	1	0	4
2	CS2251	Database Management Systems	PC	3	3	0	0	3
3	IT2251	Data Structures	PC	3	3	0	0	3
4	IT2252	Operating Systems	PC	3	3	0	0	3
5	IT2253	Object Oriented Analysis and Design	PC	3	3	0	0	3
6		Audit Course	AUD	3	3	0	0	0
ONLINE COURSE								
7	OL1	Online Course I	OL	2	2	0	0	2
PRACTICALS								
8	CS2252	Database Management Systems Laboratory	PC	4	0	0	4	2
9	IT2254	Data Structures Laboratory	PC	4	0	0	4	2
10	IT2255	Operating Systems Laboratory	PC	4	0	0	4	2
11		Internship / Value Added Course	EEC	2	0	0	2	2
			TOTAL	**35**	**20**	**1**	**14**	**26**

SEMESTER III

Course Code	Course Name	L	T	P	C
MA2201	Linear Algebra and Boundary Value Problems	3	1	0	4

Category: Foundation Courses (Basic Science Courses) (Common to ADS, CSE, IT)

a. Preamble

The operations of addition and scalar multiplication are used in many diverse contexts in mathematics. The general theory of mathematical systems involving addition and scalar multiplication has the applications to many areas of Engineering. Mathematical systems of this form are called Vector spaces or linear spaces. Subject to certain given conditions, called boundary conditions, solving partial differential equation is known as solving a boundary value problem. It is also applied in many Engineering field.

b. Course Outcomes

After successful completion of the course, the students will be able to

CO. No.	Course Outcome	Knowledge Level
CO1	Test the given system of equation is linearly dependent or independent.	K3
CO2	Apply the concept of eigenvalues and eigen vectors for diagonalization of a matrix.	K3
CO3	Apply the inner product techniques for finding the orthonormal vector and minimal solution to the system of linear equation.	K3
CO4	Apply the Fourier series techniques in solving wave and heat flow equations.	K3
CO5	Solve Initial and Boundary value problems numerically.	K3

c. Course Syllabus **Total: 60 Periods**

VECTOR SPACES **12**

Vector spaces – Subspaces – Linear combinations of vectors – Linear Span – Linear independence and linear dependence – Bases and dimensions.

LINEAR TRANSFORMATION AND DIAGONALIZATION **12**

Linear transformation – Null space and range space – Dimension theorem (proof excluded) – Matrix representation of a linear transformation – Eigen values and eigen vectors – Diagonalization of linear transformation – Applications.

INNER PRODUCT SPACES **12**

Inner products spaces – Orthogonal vectors – Gram Schmidt orthogonalization process (proof excluded) – Orthogonal complement – Least square approximation – Minimal solution to system of linear equations.

FOURIER SERIES AND BOUNDARY VALUE PROBLEMS **12**

Dirichlet's conditions – General Fourier series – Half range sine series – Half range cosine series – Classification of Partial differential equations – Fourier Series Solutions of one dimensional wave equation – One dimensional equation of heat conduction.

NUMERICAL SOLUTION OF BOUNDARY VALUE PROBLEMS **12**

Difference operators – Finite difference solution of second order ordinary differential equation – Finite difference techniques for the solution of two dimensional Laplace's and Poisson's equations on rectangular domain – Numerical solution of Parabolic: Crank Nicholson and Bender Schmidt method.

d. Activities: Students shall be exposed to MATLAB programming to solve simple problems in Diagonalization.

e. Learning Resources

i. TEXT BOOKS

1. Friedberg, A.H, Insel, A, J, & Spence, L, "*Linear Algebra*", Prentice Hall of India,
   New Delhi, 2004.
2. Grewal, B.S, "*Higher Engineering Mathematics*", Khanna Publishers, Forty Fourth Edition New Delhi, 2020.
3. Grewal, B.S, and Grewal, J.S, "*Numerical Methods in Engineering and Science*", Khanna Publishers, Tenth Edition, New Delhi, 2016.

ii. REFERENCE BOOKS

1. Kolman, B, & Hill, D.R, "*Introductory Linear Algebra*", Pearson Education, New Delhi, First Reprint, 2009.
2. Strang, G, "*Linear Algebra and its applications*", Fourth Edition, Cengage learning India pvt. Ltd, 2012.
3. Glyn James, "*Advanced Modern Engineering Mathematics*", Pearson Education, Fourth Edition, New Delhi, 2011.
4. Peter ,V, O'Neil, "*Advanced Engineering Mathematics*", Cengage Learning India Pvt., Ltd., Seventh Edition, New Delhi, 2012.
5. Kandasamy, P, Thilagavathy, K, & Gunavathy, K, "*Numerical Methods*", Third Edition Reprint, S. Chand & Co. Ltd., New Delhi, 2014.

Course Code	Course Name	L	T	P	C
IT2201	Computer Organization And Architecture	3	0	0	3

Category: Professional Core (Common to ADS, CSE, IT)

a. Preamble

This course enables the students to understand the basic structure, operations and instructions of a digital computer. This course helps the students to learn the implementation of fixed point and floating-point arithmetic operations. This course makes students familiar with the basic processing unit and multiple functional units in a processor. This course enables the students to understand the hierarchical memory system and I/O organization. This course focuses the concepts of instruction level parallelism, data level parallelism and loop level parallelism.

b. Course Outcomes

After successful completion of the course, the students will be able to:

CO. No.	Course Outcome	Knowledge Level
CO1	Summarize the functionalities of various parts, instruction sets and operations of a digital computer	K2
CO2	Utilize the logic design for fixed-point and floating point arithmetic	K3
CO3	Interpret the role of a processing unit and multiple functional units.	K3
CO4	Explain the various elements in memory hierarchy and the basic and complex I/O structures.	K2
CO5	Demonstrate how parallelism is used at instruction-level and data-level parallelism.	K2

c. Course Syllabus **Total: 45 periods**

BASIC STRUCTURE OF COMPUTERS **9**

Functional Units – Basic Operational Concepts – Bus Structures – Software – Performance: Processor Clock, Basic Performance Equation, Clock Rate – Instruction Set: CISC and RISC – Memory Locations and Addresses – Memory Operations – Instructions and Instruction Sequencing – Addressing Modes – Basic Input/output Operations.

ARITHMETIC UNIT **9**

Addition and Subtraction of Signed Numbers – Design of Fast Adders – Multiplication of Positive Numbers – Signed Operand Multiplication – Fast Multiplication – Integer Division – Floating Point Numbers and Operations.

PROCESSING UNIT **9**

Basic Processing Unit: Fundamental Concepts – Execution of a complete instruction – Multiple-bus organization – Hardwired Control – Microprogrammed control – Pipelining: Basic Concepts – Data Hazards – Instruction Hazards – Datapath and Control Considerations.

MEMORY SYSTEMS & INPUT / OUTPUT ORGANIZATION **9**

Memory Systems: Basic Concepts – Cache Memories – Performance Considerations – Virtual Memories – Memory Management Requirements – Secondary Storage – Input / Output Organization: Accessing I/O Devices – Interrupts – Direct Memory Access – Buses – Synchronous Bus – Asynchronous Bus.

PARALLEL PROCESSING **9**

Instruction-Level Parallelism: Concepts and Challenges – Basic compiler techniques for exposing ILP – Overcoming Data Hazards with Dynamic Scheduling – Dynamic Scheduling: Examples and the Algorithm – Data-Level Parallelism: Introduction – Vector Architecture – Graphics Processing Units – Detecting and Enhancing Loop-Level Parallelism.

d. Activities

Students shall be involved in various activities to improve conceptual learning:

i. Solving problems

ii. Chart work

iii. Quiz

e. Learning Resources

i) TEXT BOOK

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky and Naraig Manjikian, "*Computer Organization and Embedded Systems*", Sixth Edition, Tata McGraw Hill, 2012.

ii) REFERENCES

1. David A. Patterson and John L. Hennessy, "*Computer Organization and Design: The Hardware/Software Interface*", Fifth Edition, Morgan Kaufmann / Elsevier, 2014.
2. William Stallings, "*Computer Organization and Architecture – Designing for Performance*", Eighth Edition, Pearson Education, 2010.
3. John P. Hayes, "*Computer Architecture and Organization*", Third Edition, Tata McGraw Hill, 2012.

Course Code	Course Name	L	T	P	C
IT2202	Object Oriented Programming	3	0	0	3

Category: Professional Core

a. Preamble:

This course enables the students to understand Object Oriented Programming concepts and basic characteristics of JAVA. This course enhances programming skill using inheritance and interfaces. This course enables the students to use exception handlers and generic programming for developing JAVA applications. This course helps the students to build a JAVA applications using event driven programming, I/O streams and multithreading programming

b. Course Outcomes

After successful completion of the course, the students will be able to:

Co. No.	Course Outcomes	Knowledge Level
CO1	Demonstrate the basic concepts of object oriented programming using JAVA	K2
CO2	Make use of the OOP concept and non-access modifiers to solve real world problems	K3
CO3	Choose an appropriate exception handler and generic data type for writing a JAVA application	K3
CO4	Select the appropriate features of event driven programming and I/O streams to give solution to real time problems	K3
CO5	Apply multithreading programming to generate synchronized threads	K3

c. Course Syllabus **Total: 45 periods**

INTRODUCTION TO OBJECT ORIENTED CONCEPTS AND JAVA PROGRAMMING **10**

Introduction to Object Oriented Programming: Abstraction, Objects and Classes, Encapsulation, Inheritance, Polymorphism – **Introduction to JAVA:** Characteristics of Java, The Java Environment, Java Source File Structure, Compilation – **Fundamental Programming Structures in Java:** Data type and Variables, Operators, Decision making and Looping – **Classes:** Predefined class, User defined class, Access modifiers – **Object:** Object reference, Object cloning, Reflection –

Methods: Types of method definition – Arrays – Strings – **Constructor:** Default constructor, Parameterized constructor – **Package:** Predefined package, util package, Understanding class path, User defined package – Javadoc comments.

INHERITANCE AND POLYMORPHISM 9

Inheritance: Single Inheritance, Multilevel Inheritance, Hierarchical Inheritance, Super keyword – Interface – **Polymorphism:** Method overloading, Method overriding – **Non-Access modifiers:** Abstract class and method, Static keyword, Final keyword – **Inner class:** Nested classes, Static inner class, Anonymous class.

EXCEPTION HANDLING AND GENERIC PROGRAMMING 9

Exception Handling: Garbage collection, Finalize() method, Throwable interface, Types of exception, **Exception handlers:** Try, Catch, Finally, Throw, Throws, User define exception – **Generic programming:** Generic class, Generic method, Restrictions and limitations, Inheritance rule for generic types, Wild card types, Reflections and generics – Collection framework: Map/List, Set, Array List / Linked List, Hash Set Collection Classes, TreeMap – Lambda expression.

STREAMS AND EVENT DRIVEN PROGRAMMING 10

Input and Output: Byte stream, Character stream, Reading and writing from console and files, Object Streams and Serialization – **Java Database Connectivity (JDBC):** Creating a database, Insertion operation, Deletion operation, Updation operation, Display operation – Event Driven programming: Introduction to Swing, MVC Framework, Frame, **Components:** Text field, Input, Choice, Text Area, Buttons, Checkboxes, Radio Buttons, Lists, Menus, Dialog Box, Windows, Mouse, **Layout Management:** Border layout, Flow layout, Card layout, Grid layout, Gridbag layout – **Listeners:** ActionListener, ItemListener, MouseListener, KeyboardListener, WindowListener – Adapter classes.

MULTITHREADING PROGRAMMING 7

Multithreading: Thread states, Thread life cycle, Thread properties, Thread priorities, Thread synchronization – Archive – Case study.

d. Activities:

Students shall be involved in various activities to improve conceptual learning:

- Code debugging
- Find the output
- Code building

e. Learning Resources

i. TEXT BOOKS

1. Cay S. Horstmann, Gary Cornell, "*Core Java: Volume I – Fundamentals*", Prentice Hall, Tenth Edition, 2015.
2. Cay S. Horstmann, Gary Cornell, "*Core Java: Volume II – Fundamentals*", Prentice Hall, Tenth Edition, 2016.

ii. **REFERENCE BOOKS**

1. Herbert Schildt, "*Java: The Complete Reference*", Eleventh Edition, McGraw Hill Education, 2014.
2. Paul Deitel ,Harvey Deitel " *Java SE8 for Programmers*", Pearson Education, Third Edition,2014.
3. P.J.Deitel&H.M.Deitel, "*Java: How to Program Java 2*", Prentice Hall, Seventh Edition, 2011.

Course Code	Course Name	L	T	P	C
IT2203	Software Engineering	3	0	0	3

a. Preamble:

This course enables the students to understand the phases in a software project development and learn how to elicit and formulate requirements. This course helps students to understand the various software design methodologies and learn various testing and maintenance measures. This course familiarize the activities in software project management

b. Course Outcomes

After successful completion of the course, the students will be able to:

Co. No.	Course Outcomes	Knowledge Level
CO1	Compare and contrast the various Process Models to develop software projects.	K2
CO2	Explain the concepts of requirement engineering and analysis modelling.	K2
CO3	Illustrate the software design process and various types of design models.	K2
CO4	Paraphrase the relevant coding standards, testing practices and Reengineering Process Model.	K2
CO5	Outline the various activities involved in the software project management.	K2

c. Course Syllabus **Total: 45 periods**

SOFTWARE PROCESS MODELS **9**

Introduction to software engineering – Software Process – Perspective Process Models: Waterfall Model – Incremental Process Models – Evolutionary Process Model – Concurrent Models. Specialized Process Models: Component-Based Development – Formal Methods Model – Aspect-Oriented Software Development. Agile Process Model: Introduction to Agility – Agile Process – Agile Manifesto and Principles – Extreme Programming – Scrum Process.

REQUIREMENT ANALYSIS AND SPECIFICATION **9**

Software Requirements: Functional Requirements – Non-Functional Requirements – User Requirements – System Requirements – Software Requirements Document. Requirement Engineering Process: Feasibility Studies – Requirements Elicitation and Analysis – Requirements Validation – Requirements Management – Requirements Modelling – Data Dictionary.

SOFTWARE DESIGN **9**

Design Process – Design Concepts – Design Model – Design Heuristic – Architectural Design: Architectural styles – Architectural Design – Mapping Data Flow and Transaction Flow into Software Architecture. User Interface Design: Interface Analysis – Interface Design. Component-Level Design: Designing Class-Based Components.

TESTING AND MAINTENANCE **9**

Taxonomy of Software Testing – Types of Testing: Black Box Testing – White Box Testing – Regression Testing – Unit Testing – Integration Testing – Validation Testing – System Testing – Debugging. Software Implementation Techniques: Coding Practices – Refactoring. Maintenance and Reengineering: BPR model – Reengineering Process Model – Reverse and Forward Engineering – Testing Tools.

SOFTWARE PROJECT MANAGEMENT **9**

Estimation: LOC – FP Based Estimation – Make/Buy Decision – COCOMO I & II Model. Project Scheduling: Project Scheduling – Scheduling – Earned Value Analysis. Project Planning: Project Plan – Planning Process. Risk Management: Software Risks – Risk Identification – Risk Projection – RMMM – RMMM Plan. Software Configuration Management: SCM Repository – SCM Process – CASE TOOLS.

d. Activities:

Students shall be involved in various activities to improve conceptual learning:

- Chart work
- Quiz

e. Learning Resources

i. TEXT BOOKS

1. Roger S. Pressman, "*Software Engineering – A Practitioner's Approach*", Eightth Edition, Mc Graw-Hill International Edition, 2015.
2. Ian Sommerville, "*Software Engineering*", 9th Edition, Pearson Education Asia, 2011.

ii. REFERENCES

1. Rajib Mall, "*Fundamentals of Software Engineering*", Third Edition, PHI Learning. PrivateLimited, 2009.
2. Pankaj Jalote, "*Software Engineering, A Precise Approach*", Wiley India, 2010.
3. Kelkar S.A., "*Software Engineering*", Prentice Hall of India Pvt Ltd, 2007.
4. Stephen R.Schach, "*Software Engineering*", Tata McGraw-Hill Publishing Company Limited, 2007.

Course Code	Course Name	L	T	P	C
EC2203	Digital Systems	3	0	0	3

Category: Engineering Sciences Courses

a. Preamble

This course promotes students to understand basic concept of combinational and sequential circuits in digital circuit. It also enables the student to understand this course promotes students to understand basic concepts of logic families and programmable devices.

b. Course Outcomes

After successful completion of the course, the students will be able to

CO. No.	Course Outcome	Knowledge Level
CO1	Outline the Boolean functions and various minimization techniques.	K2
CO2	Illustrate the combinational circuits used to perform basic digital operations	K2
CO3	Develop a synchronous/asynchronous counters and shift registers using sequential logic.	K3
CO4	Implement combinational and sequential logic circuits using Verilog HDL.	K3
CO5	Design combinational circuits using programmable logic devices and Memory Devices.	K3

c. Course Syllabus **Total: 45 Periods**

DIGITAL FUNDAMENTALS **9**

Review of Number systems, Logic gates, Boolean algebra, Boolean postulates and laws - De-Morgan's Theorem - Principle of Duality, Simplification using Boolean algebra, Canonical forms - Sum of product and Product of sum - Minimization using Karnaugh map - NAND and NOR Implementation

COMBINATIONAL CIRCUITS **9**

Realization of combinational logic using gates , Design of combinational circuits : Adder , Subtractor, Parallel adder / Subtractor, Magnitude Comparator, Parity

generator and checker, Encoder, Decoder, Multiplexer, Demultiplexer - Code converters.

SYNCHRONOUS SEQUENTIAL CIRCUITS 9

Latches, Flip-Flops-SR, JK, D & T, Shift Registers - SISO, SIPO, PISO, PIPO, Design of Synchronous Sequential Circuits - State Table and State Diagrams, Design of Counters - Modulo N counters, Random Sequence counters, Johnson counter, Ring counter, Up/Down counters.

MODELLING OF LOGIC CIRCUITS BY VERILOG HDL 9

Lexical Conventions, Ports and Modules, Gate Level Modelling, Operators, Data Flow Modelling, Behavioral level Modelling - Modelling of Combinational and Sequential Logic Circuits using Verilog HDL.

MEMORY AND PROGRAMMABLE LOGIC DEVICES 9

RAM - ROM - Basic Structure, Types - Programmable Logic Devices – Programmable Logic Array (PLA) - Programmable Array Logic (PAL) – Implementation of combinational logic circuits using PLA, PAL. Hazards - Hazard free realization.

d. Activities: Students shall be given exposure to understand the combinational and sequential digital circuits and to develop an application by using digital logic.

e. Learning Resources

i. **TEXT BOOK**

1. M. Morris Mano, Michael D. Ciletti,2017, *Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog*, 6th Edition, Pearson Education.and Synthesis, McGraw Hill

ii. **REFERENCE BOOKS**

1. Charles H.Roth. 2013 *Fundamentals of Logic Design*, 6th Edition, Thomson Learning.
2. Wakerly J F, 2002, *Digital Design: Principles and Practices*, Prentice-Hall, 2nd Ed.
3. D. D. Givone, 2003, *Digital Principles and Design*, Tata Mc-Graw Hill, New Delhi.
4. Thomas L. Floyd, 2011, *Digital Fundamentals*, 10th Edition, Pearson Education Inc.

5. Stephen Brown and ZvonkoVranesic, 2013, *Fundamentals of Digital Logic with Verilog Desig*n, Third Edition, McGraw-Hill Higher Education, New Delhi, India.

Course Code	Course Name	L	T	P	C
EE2201	Fundamentals of Electrical and Electronics Engineering	3	0	0	3

Category: Engineering Sciences Courses

a. Preamble

This course introduces the concepts related to basic of electric circuits, analog and digital electronics, electrical machines and different measuring instruments.

b. Course Outcome

After successful completion of the course, the students will be able to

CO. No.	Course Outcome	Knowledge Level
CO1	Solve simple dc circuits using basic electrical laws	K2
CO2	Describe the construction and working principle of various DC and AC Machines	K2
CO3	Elucidate characteristics of various semiconductor devices used in electronic circuits	K3
CO4	Design simple digital circuits for various electronic applications	K3
CO5	Explain the construction and working of electrical measuring instruments and transducers	K2

c. Course SyllabusTotal: 45 Periods

BASIC ELECTRICAL CIRCUITS AND SYSTEMS 9

Electrical circuit elements (R, L and C) - Dependent and independent DC sources – Ohm's Law- Kirchhoff's laws - mesh current and node voltage methods (Analysis with only independent DC source) - Three phase supply (Star & Delta connection) - Basics of Energy Tariff calculation.

ELECTRICAL MACHINES 9

Principles of operation and characteristics of; DC machines (Series, Shunt Motors and Generator) - Construction and working of Transformers (single and three phase), AC generators - single phase capacitor start/run induction motors.

ELECTRONIC DEVICES & CIRCUITS **9**

Types of Materials – Silicon & Germanium- N type and P type materials – Operation & VI characteristics; PN Junction Diode & Zener Diode – Bipolar Junction Transistor (Characteristics & Biasing) – Introduction to Operational Amplifier –Inverting Amplifier –Non Inverting Amplifier – Passive Filters (Low pass & High Pass).

DIGITAL ELECTRONICS **9**

Number System – Basic Boolean laws – Demorgan's theorem– Logic Gates - Introduction to combinational Circuits (Half adder, Full adder, Multiplexer and Demultiplexer) - Introduction to sequential Circuits (SR, JK, D, T Flip-Flops - Registers and Modulo Counters).

MEASUREMENTS & INSTRUMENTATION **9**

Classification of instruments - Types of indicating Instruments – Construction and working; Induction Wattmeter, Ammeter (moving coil and moving iron type), Voltmeter (moving coil and moving iron type) – CRO – three-phase power measurements (Two wattmeter method) - Introduction to transducers (LVDT, RTD and Piezoelectric).

d. Activities

Students shall be exposed to the basic electric engineering projects.

e. Learning Resources

i. TEXT BOOKS

1. Muthusubramanian, R., Salivahanan, R. and Muraleedharan, K.A., 2009. *Basic Electrical & Electronics Engineering*. Tata McGraw Hill Education Private Limited.

ii. REFERENCE BOOKS

1. Theraja, B.L., 2006. *Fundamentals of Electrical Engineering and Electronics in SI System of Units (including Rationalized MKSA System)*. Chand.
2. Bird, J., 2014. *Electrical circuit theory and technology*. Routledge.
3. Al Morris, A.S. and Langari, R., 2012. *Measurement and instrumentation: theory and application*. Academic Press.
4. Prasad, R., 2014. *Fundamentals of Electrical Engineering*. PHI Learning Pvt. Ltd.

Course Code	Course Name	L	T	P	C
GE2201	Design Thinking	3	0	0	3

Category: Employability Enhancement Course

a. Preamble

This course introduces the various principles of design thinking to achieve an effective design and to examine the implementation of the model or process for its successful operation.

b. Course Outcomes

After successful completion of the course, the students will be able to

CO. No.	Course Outcome	Knowledge Level
CO1	Describe the basic principles of design and various stages of design thinking for better conceiving of idea and refinement	K2
CO2	Elucidate the concepts of idea generation and refinement	K3
CO3	Apply various prototype models for solving complex problems	K3
CO4	Analyze real-time problems for effective design, implementation and operation	K3
CO5	Device idea/solution towards development of a prototype for a chosen problem of interest	K4

c. Course Syllabus **Total: 45 Periods**

INTRODUCTION TO DESIGN THINKING **9**

Introduction - Product life cycle – Design Ethics – Design Process – Stages in design thinking: Immersion, Analysis and synthesis, Ideation, Prototyping.

IDEA GENERATION AND REFINEMENT **9**

Basic design - directions - Themes of thinking - Inspiration and references - Brainstorming - Value - Inclusion – Sketching - Presenting ideas - Thinking in images - Thinking in signs - Appropriation - Personification - Visual metaphors - Modification - Thinking in words – Words and language - Thinking in shapes - Thinking in proportions - Thinking in color - Outside the Box.

PROTOTYPING **9**

Developing designs - Types of prototype - Prototyping for Designing Complex Systems – The Efficacy of Prototyping under Time Constraints.

IMPLEMENTATION **9**

Format - Materials - Finishing - Media - Scale - Series/Continuity - Emerging Landscapes of Design - Real-Time Design Interaction Capture and Analysis - Enabling Efficient Collaboration in Digital Design - Spaces Across Time and Distance - Software used in Developing in Virtual Environments.

DESIGN THINKING IN VARIOUS SECTORS **9**

Design & Development of Prototypes for Wall Plastering, Rubber shredding, Separation of Corn seeds, Electric vehicles, Smart gates, Burglar alarm, Tyre pressure monitor, Development of Online Voting System, Online Proctoring System, Online Health Monitoring System, IoT based Home Automation and any other problem of interest in your domain.

d. Learning Resources

i. TEXT BOOKS:

1. Binder, T., De Michelis, G., Ehn, P., Jacucci, G., Linde, P., and Wagner, I., 2011, *Design things*, MIT press

2. Ambrose, G., and Harris, P., 2009. *Basics Design: Design thinking*, Bloomsbury Publishing

ii. REFERENCE BOOKS:

1. Meinel, C., and Leifer, L. (Eds.)., 2011. *Understanding Innovation*, Springer.
2. Plattner, H., Meinel, C., and Leifer, L. (Eds.)., 2010. *Design thinking: understand– improve–apply*, Springer Science & Business Media
3. Moran, T. P., and Carroll, J. M., 1996. *Design Rationale: Concepts, Techniques, and Use*, L. Erlbaum Associates Inc.
4. Cross, N., 1984. *Developments in Design Methodology, Chichester*: Wiley.

WEB RESOURCES:

1. https://www.designsociety.org/download-publication/39626/Design+prototyping+of+systems

2. https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process

VIDEO LECTURES :(NPTEL OR ANY OTHER VIDEO LECTURES)

1. https://nptel.ac.in/courses/110/106/110106124/#

Course Code	Course Name	L	T	P	C
IT2204	Object Oriented Programming Laboratory	0	0	4	2

Category: Professional Core Course

a. **Preamble**

This course enables the students to build software development skills using JAVA programming for real-world applications and understand and apply the OOPs concepts like inheritance, interfaces. This course enables the students to handle the exceptions that arise in JAVA applications and apply the concepts of event driven programming and JDBC to store and retrieve data from database. This course enables the students to develop applications using generic programming and multithreading

b. Course Outcomes

After successful completion of the course, the students will be able to:

Co. No.	Course Outcomes	Knowledge Level
CO1	Develop JAVA applications using Fundamental Programming Structures	K3
CO2	Make use of the OOPs features to implement various JAVA applications	K3
CO3	Apply the exception handling mechanism to handle the exceptions that arise in JAVA applications	K3
CO4	Build Java application using event driven programming and JDBC concepts	K3
CO5	Utilize Generics programming and Multithreaded programming for developing JAVA applications	K3

c. Course Syllabus **Total: 45 periods**

LIST OF EXPERIMENTS:

1. **Implementation of Basic Java programs**
    a. Make use of appropriate control statements
        i. To perform linear search
        ii. To perform matrix operations
        iii. To generate prime numbers
        iv. Pattern printing – Floyd's triangle
    b. Build a user defined classes and object
    c. Develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial). Compute the bill amount using the following tariff.
        i. If the type of the EB connection is domestic, calculate the amount to be paid as follows:
            - First 100 units – Rs. 1 per unit
            - 101-200 units – Rs. 2.50 per unit
            - 201 -500 units – Rs. 4 per unit
            - > 501 units – Rs. 6 per unit
        ii. If the type of the EB connection is commercial, calculate the amount to be paid as follows:
            - First 100 units – Rs. 2 per unit
            - 101-200 units – Rs. 4.50 per unit
            - 201 -500 units – Rs. 6 per unit
            - > 501 units – Rs. 7 per unit
2. **Implementation of user defined Packages.**
    a. Develop a Java application to implement currency converter (Dollar to INR, EURO to INR, Yen to INR and vice versa), distance converter (meter to KM, miles to KM and vice versa) , time converter (hours to minutes, seconds and vice versa) using packages.
3. **Implementation of Inheritance concepts**

a. Develop a Java application with Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.

4. **Implementation of Interfaces concept**
   a. Design a Java interface for ADT Stack. Implement this interface using array. Provide necessary exception handling in both the implementations.
5. **Implement a Java program that make use of Non access modifiers**
   a. Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area().
      - Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape.
      - Each one of the classes contains only the method print Area () that prints the area of the given shape.
6. **Implement a Java program using various Exception handling**
   a. Write a Java program to use exception handlers
7. **Files and IO streams.**
   a. Write a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.
8. **Implementation of JDBC**
   a. Make use of Q1.c. Create a table that contains cid, cname, address, date_of_payment, previous_month_reading, current_month_reading, connection_type, amount. Perform the following operations:
      i. Insert the electricity meter reading units details of customer
      ii. Display the details of the customer of the type Domestic and commercial
      iii. Update the address of the specific customer
      iv. Delete the details of a specific customer
      v. Display the total bill amount generated in a specific duration

vi. Display the details of the customer who is paying highest amount

9. Implement a real time application using Event driven program

a. Design a calculator using Event-driven programming paradigm of Java with the following options.

i. Decimal manipulations

ii. Scientific manipulations

10. Implementation of Generics programming

a. Write a Java program to find the maximum value from the given type of elements using a generic function.

11. Utilize the appropriate Collection framework for any real time application

a. Write a program to perform string operations using ArrayList. Write functions for the following

i. Append

ii. Insert

iii. Search

iv. List all string starts with given letter

12. Implementation of Multithreading programming

a. Write a java program that implements a multi-threaded application that has three threads.

i. First thread generates a random integer every 1 second and if the value is even

ii. Second thread computes the square of the number and prints. If the value is odd

iii. Third thread will print the value of cube of the number.

13. Mini project

TOTAL: 60 PERIODS

SOFTWARE SPECIFICATIONS:

1. JDK8
2. Eclipse / Netbean
3. MySQL

d. Learning Resources

REFERENCES

1 Herbert Schildt, "*Java: The Complete Reference*", Eleventh Edition, McGraw Hill Education, 2014.

2 https://www.eclipse.org/

3 https://netbeans.org/

Course Code	Course Name	L	T	P	C
EC2204	Digital Systems Laboratory	0	0	4	2

Category: Professional Core Course

a. Preamble

This course promotes students to design circuits for different digital systems in Engineering.

b. Course Outcomes

After successful completion of the course, the students will be able to

CO No.	Course Outcome	Knowledge Level
CO1	Experiment with the basics of gates.	K3
CO2	Build different combinational circuits.	K3
CO3	Construct various sequential circuits.	K3
CO4	Model combinational & Sequential circuits using HDL.	K3
CO5	Make use of the concepts for implementation of a simple digital system.	K3

c. Course Syllabus **Total: 60 Periods**

Design and Analysis of the following Circuits

1. Verification of Boolean Theorems using basic gates.

2. Design and implementation of combinational circuits using basic gates for arbitrary functions, code converters.

3. Design and implement Half/Full Adder and Subtractor.

4. Design and implement combinational circuits using MSI devices:

- 4–bit binary adder / subtractor
- Parity generator / checker
- Magnitude Comparator
- Application using multiplexers and demultiplexers.

5. Design and implement shift-registers.

6. Design and implement synchronous counters.

7. Design and implement asynchronous counters.

8. Coding combinational circuits using HDL.

9. Coding sequential circuits using HDL.

10. Design and implementation of a simple digital system (Mini Project).

d. Activities

Students shall be given exposure to design a circuit and analyse its performance through hardware components and the same using HDL. Based on the gained knowledge, insisted them to do mini projects.

e. Learning Resources

i. **TEXT BOOKS**

1. M. Morris Mano and Michael D. Ciletti, 2014, *Digital Design*, 5th Edition, Pearson.
2. Samir Palnitkar, 1996, Verilog HDL: A Guide to Digital Design and Synthesis

ii. **REFERENCE BOOKS**

5. Charles H.Roth, 2013, *Fundamentals of Logic Design*, 6th Edition, Thomson Learning.
6. Thomas L. Floyd, 2011, *Digital Fundamentals*, 10th Edition, Pearson Education Inc.

7. S.Salivahanan and S.Arivazhagan, 2012, *Digital Electronics*, 1st Edition, Vikas Publishing House pvt Ltd.
8. Anil K.Maini, 2014, *Digital Electronics*, Wiley.
9. A.Anand Kumar, 2016, *Fundamentals of Digital Circuits*, 4th Edition, PHI Learning Private Limited.
10. Soumitra Kumar Mandal, 2016, *Digital Electronics*, McGraw Hill Education Private Limited.

SEMESTER IV

Course Code	Course Name	L	T	P	C
MA2251	Discrete Mathematics and Probability	3	1	0	4

Category: Foundation Courses (Basic Science Courses) (Common to CSE, IT)

a. Preamble

A course in discrete mathematics represents the discrete structures like predicate logic and proposition. The general counting methods involve permutations and combinations are very useful in constructing computer programs and in mastering many theoretical topics of computer science. The probability theory gives adequate exposure in random variables, probability distributions, regression and correlation.

b. Course Outcomes

After successful completion of the course, the students will be able to

CO. No.	Course Outcome	Knowledge Level
CO1	Use propositional and predicate logic to derive new inference from a given scenario.	K3
CO2	Solve problems using mathematical induction, permutation, combination and recurrence relations.	K3
CO3	Apply graph theory to find shortest path and Euler's circuits in a given network.	K3
CO4	Apply the concepts of probability distributions to solve engineering problems.	K3
CO5	Compute the correlation between two random variables and linear regression equation for a given set of data.	K3

c. **Course Syllabus** **Total: 60 Periods**

PROPOSITION AND PREDICATE LOGIC **12**

Basic connectives – Truth Table – Tautological Implications – Propositional equivalences – Normal Forms – Rules of inference – Predicates and quantifiers – Nested quantifiers – Universe of discourse – Theory of inference for Predicate calculus.

COMBINATORICS **12**

Mathematical induction – Strong induction and well ordering – The basics of counting – The pigeonhole principle – Permutations and combinations – Recurrence relations – Solving linear recurrence relations – Generating functions – principle of Inclusion and exclusion and its applications.

GRAPH THEORY AND ITS APPLICATIONS **12**

Graphs – Matrix representation of graphs – Graph isomorphism – connectivity – Eulerian and Hamiltonian graphs (Proof excluded) – Prim's Algorithm – Kruskal's Algorithm – Problems.

PROBABILITY AND RANDOM VARIABLE **12**

Probability – Conditional probability – Baye's theorem – Random variables – Mathematical Expectations – Moments – Moment generating functions – Distributions: Binomial, Poisson, Geometric, Uniform, Exponential and Normal distribution.

TWO DIMENSIONAL RANDOM VARIABLES **12**

Joint distributions – Marginal and Conditional distributions – Covariance – Correlation and linear regression – Central limit theorem (proof excluded) – Transformation of Random Variables.

d. **Activities:** Solving discrete and continuous distribution problems using Electronic Spread Sheet.

e. Learning Resources

i. TEXT BOOKS

1. Rosen, K.H, "*Discrete Mathematics and its Applications*", Tata McGraw Hill, Seventh Edition, New Delhi, 2015.
2. Johnson, R. A, Miller, I & Freund J, "*Miller and Freund's Probability and Statistics for Engineers*", Pearson Education, Eighth Edition, Asia, 2015.

ii. REFERENCE BOOKS

1. Grimaldi, R.P, "*Discrete and Combinatorial Mathematics: An Applied Introduction*", Pearson Education Asia, Fifth Edition, Delhi, 2019.
2. Lipschutz, S & Mark Lipson, "*Discrete Mathematics*", Tata McGraw Hill, Third Edition, New Delhi, 2010.
3. Kenneth H Rosen, "*Discrete Mathematics with Applications*", Tata

McGraw Hill, Eighth Edition, 2021.

4. Ross, S.M, "*Introduction to Probability and Statistics for Engineers and Scientists*", Elsevier, Fifth Edition, 2014.
5. Spiegel, M.R, Schiller, J & Srinivasan, R.A, *Schaum's Outline of Theory and Problems of Probability and Statistics*, Tata Mc Graw Hill, Fourth Edition, 2012.

Course Code	Course Name	L	T	P	C
CS2251	Database Management Systems	3	0	0	3

Category: Professional Core Course (Common to IT, CSE)

a. Preamble

This course enables the students to learn the fundamentals of data models and to represent a database system using ER diagrams and study SQL and relational database design. This course enriches the knowledge in the internal storage structures using different file and indexing techniques which will help in physical database design. This course enables the students to understand the fundamental concepts of transaction processing- concurrency control techniques and recovery procedures and learn about file organization and query processing

b. COURSE OUTCOMES

After successful completion of the course, the students will be able to:

CO. No.	Course Outcome	Knowledge Level
CO1	Infer the basic concepts of database system and model ER diagram for real time applications	K3
CO2	Use appropriate SQL commands to store and access data from Relational Database.	K3
CO3	Construct normalized database for real world scenario using functional dependencies	K3
CO4	Illustrate the importance of concurrency control in transaction to maintain consistency in a database	K3
CO5	Interpret the mechanisms incorporated in file organization and Query	K3

c. Course Syllabus **Total: 45 periods**

INTRODUCTION TO DATABASE & ER MODEL **9**

Introduction to Databases - File System Vs Database System - Database System Architecture- Database Users and Administrator - Data Models - Entity Relationship Model - E-R Diagrams - Design Issues - Extended E-R Features - Introduction to Relational Model - ER to Relational Schema Mapping

RELATIONAL MODEL & SQL **9**

Structure of Relational Databases - Relational Query Languages - Relational Algebra – SQL: DDL, DML, DCL, TCL - Simple Queries, Complex Nested Queries, Correlated Nested Queries, Joins, Aggregate Functions, Grouping - PL/SQL : Functions, Procedures, Triggers, Views -Embedded SQL - Dynamic SQL

NORMALIZATION **9**

Pitfalls in Bad Relational database design - Functional Dependencies (Closure of Functional dependencies) - Closure of Attributes - Normal Forms : First, Second, Third, Boyce Codd Normal Form, Multivalued Dependencies : Fourth Normal Form, Join Dependencies : Fifth Normal Form –Domain Key Normal Form

TRANSACTION AND CONCURRENCY CONTROL **9**

Transaction processing concepts - Need for concurrency control and recovery - Recoverability – Transaction Recovery – Serializability : Conflict Serializability, View Serializability, Testing for Serializability - Concurrency Control : Lock Based Protocols (Two phase locking Techniques, Strict Two Phase Locking, Deadlocks, Multiple Granularity) Timestamp Based protocol, Validation Based protocol

FILE ORGANIZATION & QUERY PROCESSING **9**

File Organization : Organization of Records in Files, Indexing and Hashing, Ordered Indices - Query Processing: Measures of Query Cost (Selection, Sorting and Join Operation), Query Tuning, Query Optimization (Transformation of Relational Expressions, Choice of Evaluation Plans, Materialized Views) – No SQL – Mongo DB.

d. Activities

Students shall be involved in various activities to improve conceptual learning:

i. Chart preparation

ii. Quiz

e. Learning Resources

i) TEXT BOOKS

i. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, "*Database System Concepts"*, 6th edition, Tata McGraw Hill. 2017.

ii. Ramez Elmasri, Shamkant B. Navathe, *"Fundamentals of Database Systems",* 6th edition, Pearson Education, 2011.

ii) REFERENCES BOOKS

1. C.J. Date, A. Kannan & S. Swamynathan, "*An Introduction to Database Systems*", 8th edition, Pearson Education, 2006.
2. Raghu Ramakrishnan, "*Database Management Systems"*, 4th edition, McGraw-Hill College Publications, 2015.
3. G.K.Gupta, "*Database Management Systems"*, Tata McGraw Hill, 2011.

Course Code	Course Name	L	T	P	C
IT2251	Data Structures	3	0	0	3

Category: Professional Core Course

a. Preamble

This course enables the students to understand the concepts of ADTs, learn linear data structures – lists, stacks, and queues, apply Tree and Graph structures and understand sorting, searching and hashing algorithms

b. Course Outcomes

After successful completion of the course, the students will be able to:

Co. No.	Course Outcomes	Knowledge Level
CO1	Utilize an appropriate linear data structure to provide solution for real life scenario	K3
CO2	Make use of Stack and Queue ADTs for problem solving.	K3
CO3	Illustrate the structural properties and operations on various types of Tree ADTs in balanced search.	K2
CO4	Select an appropriate graph algorithm to solve real life problems.	K3
CO5	Choose an appropriate sorting, searching or indexing strategy for effective data storage and retrieval.	K3

c. Course Syllabus **Total : 45 periods**

LINEAR DATA STRUCTURES - LIST **9**

ADTs- List - Array & Singly linked list: Polynomial Manipulation, Merging of Two lists - Doubly linked list: Palindrome Checking - Circular linked list: Round Robin Scheduling, Josephus Problem.

LINEAR DATA STRUCTURES - STACKS, QUEUES **9**

Stack - Polish Form : Infix to Postfix, Evaluation of Postfix - Parenthesis Checking- Palindrome Checking- Recursion Avoidance – Queue- FIFO Scheduling- Deque- Priority Queue: Priority Based Scheduling.

NON LINEAR DATA STRUCTURES - TREES **9**

Trees - Binary Search Tree: Traversal - AVL Tree - B Tree - B+ Trees - Heap

NON LINEAR DATA STRUCTURES - GRAPHS **9**

Graphs – Traversal: BFS, DFS - Minimum Path: Dijkstra's - Spanning Tree: Prims,Kruskal - Bi-connectivity & Cut vertices - Topological Sort - Euler's Tour

SEARCHING, SORTING AND HASHING TECHNIQUES **9**

Searching: Linear Searching, Binary Searching – Sorting : Bubble Sort - Insertion Sort - Selection Sort - Shell Sort - Radix Sort - Indexing – Hashing: Closed Hash, Open hash, Collision Avoidance: Linear, Quadratic, Double Hashing – Rehashing - Extendible Hashing

d. Activities

Students shall be involved in various activities to improve conceptual learning:

i. Chart preparation

ii. Role play

iii. Quiz

e. Learning Resources

i. TEXT BOOKS

1. Horowitz & Sahni, "*Fundamentals of Data Structures in C*", 2nd edition, Orient Publication, 2008.

2. Aho, Hopcroft & Ullman, "*Data Structures and Algorithms*", Addison Wesley., 1983.

ii. REFERENCES

1.Aaron M. Tenenbaum, Yedidyah Langsam & Moshe J. Augenstein, "*Data Structures Using C and C++*", PHI Publications, 2006.

2.Jean Paul Trembley & Paul G. Sorenson, 2017, "*An Introduction to Data Structures with applications"*, 2nd edition, McGraw Hill Publications

3. Mark Allen Weiss, "*Data Structures and Algorithm Analysis in C*", 2nd edition, Addison-Wesley, 2002.

Course Code	Course Name	L	T	P	C
IT2252	Operating Systems	3	0	0	3

Category: Professional Core Course

a. Preamble

This course enables the students to acquire basic knowledge on operating system structures and its functions, study the concept of process management and deadlock, learn the basics of memory management and its techniques, understand the structure of file, Directory and I/O systems and families with some operating systems

b. Course Outcomes

After successful completion of the course, the students will be able to:

Co. No.	Course Outcomes	Knowledge Level
CO1	Elucidate the evolution of operating system along with its structure and functions	K2
CO2	Demonstrate the various process management algorithms	K2
CO3	Illustrate the performance of various memory management techniques	K2
CO4	Describe file, Directory system and I/O Management techniques	K2
CO5	Summarize some popular operating systems like Linux, Mobile OS like iOS and Android	K2

c. Course Syllabus **Total : 45 peroids**

INTRODUCTION **9**

Introduction to Operating Systems - Computer System Organization - Architecture - Evolution of Operating System - Operating System Structure - Operations - Process, Memory, Storage Management - Protection and Security - Distributed Systems - OS Services - User interface - System Calls - System Programs - Process Concept - Scheduling - Operations on Processes - Cooperating Processes - Inter process Communication - Threads

PROCESS MANAGEMENT **10**

Scheduling : Scheduling Criteria - Scheduling Algorithms - Multiple Processor Scheduling - Algorithm Evaluation - The Critical Section Problem - Synchronization Hardware - Semaphores - Classic Problems of Synchronization - Critical Regions - Monitors - Deadlocks - Deadlock Characterization - Methods for Handling Deadlocks - Deadlock Prevention - Deadlock Avoidance - Deadlock detection and Recovery

MEMORY MANAGEMENT **9**

Introduction - Swapping - Contiguous Memory Allocation - Paging - Segmentation - Segmentation with Paging - Virtual Memory: Background - Demand Paging - Page Replacement - Allocation of Frames - Thrashing

FILE AND I/O SYSTEMS **8**

File Concept - Access Methods - Directory Structure - File System Mounting - Protection - Directory Implementation - Allocation Methods - Free Space Management - Disk Scheduling - Disk Management - Swap Space Management - Protection. I/O Systems: I/O Hardware - Application I/O Interface - Kernel I/O Subsystem.

CASE STUDY **9**

The Linux System: History - Design Principles - Kernel Modules - Process Management Scheduling - Memory management - File systems - Input and output - Inter Process Communication - Mobile OS - iOS and Android - Architecture and SDK Framework, Media Layer, Services Layer, Core OS Layer, File System.

d. Activities

Students shall be involved in various activities to improve conceptual learning:

i. Role play

ii. Quiz

iii. Case study presentation

e. Learning Resources

i. TEXTBOOK:

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, "*Operating System Concepts*", John Wiley & Sons Inc., 9th Edition, 2013

ii. REFERENCE BOOKS:

1. Andrew S. Tanenbaum, "*Modern Operating Systems*", Addison Wesley, Second Edition, 2001.

2. William Stallings, "*Operating Systems: Internals and Design Principles*", Prentice Hall, Seventh Edition, 2011.

3. Charles Crowley, "*Operating Systems: A Design-Oriented Approach*", Tata McGraw Hill Education, 1996.

4. D M Dhamdhere, "*Operating Systems: A Concept-based Approach*", Tata McGraw-Hill Education, Second Edition, 2007.

5. Neil Smyth, "*iPhone iOS 4 Development Essentials – Xcode*", Fourth Edition, Payload media, 2011.

6. Daniel P Bovet and Marco Cesati, "*Understanding the Linux kernel*", 3rd edition, O'Reilly, 2005.

Course Code	Course Name	L	T	P	C
IT2253	Object Oriented Analysis and Design	3	0	0	3

Category: Professional Core Course

INTRODUCTION & CASE STUDY - NEXTGEN POS SYSTEM 9

Introduction to OOAD with OO Basics - Unified ProcessModel- Case study - The Next Gen POS system - Inception - Use case Modelling - Relating Use cases – include, extend and generalization

Case Study:

- Draw standard UML diagrams using an UML modeling tool for a given case study and map design to code and implement a 3 layered architecture.

- Test the developed code and validate whether the SRS is satisfied.

1. Identify a software system that needs to be developed
2. Document the Software Requirements Specification (SRS) for the identified system.
3. Identify use cases and develop the Use Case model.

STATIC UML DIAGRAMS **9**

Class Diagram - Elaboration - Domain Model - Finding conceptual classes and description classes - Associations - Attributes - Domain model refinement - Finding conceptual class Hierarchies - Aggregation and Composition - Relationship between sequence diagrams and use cases

DYNAMIC AND IMPLEMENTATION DIAGRAMS **9**

Dynamic Diagrams -UML interaction diagrams -System sequence diagram - Collaboration diagram -State machine diagram and Modelling -Activity diagram - UML package diagram - Component and Deployment Diagrams

DESIGN PATTERNS AND MAPPING DESIGN TO CODE **9**

GRASP: Designing objects with responsibilities - Creator - Information expert Low Coupling - High Cohesion - Controller Design Patterns - creational - factory method - structural - Bridge - Adapter - behavioural - Strategy – Observer - Applying GoF design patterns - Mapping design to code

OBJECT ORIENTED TESTING **9**

Impact of object orientation on Testing - Testing: Issues in OO Testing - Class Testing - OO Integration Testing - GUI Testing - OO System Testing. - Develop Test Cases and Test Plans

d. Activities

Students shall be involved in various activities to improve conceptual learning:

i. Chart presentation

ii. Quiz

iii. Case study presentation

e. Learning Resources

i. TEXT BOOKS

1. Craig Larman, *Applying UML and Patterns: An Introduction to object-oriented Analysis and Design and iterative development*, Third Edition, Pearson Education, 2012.

2. Paul C. Jorgensen, *Software Testing:- A Craftsman"s Approach, Third Edition, Auerbach Publications*, Taylor and Francis Group, 2008.

ii. REFERENCES

1. Mike O'Docherty, *Object-Oriented Analysis & Design: Understanding System Development with UML 2.0*, John Wiley & Sons, 2005.

2. James W- Cooper, Addison-Wesley, *Java Design Patterns – A Tutorial*, 2000.

3. MichealBlaha, James Rambaugh, *Object-Oriented Modeling and Design with UML*, Second Edition, Prentice Hall of India Private Limited, 2007.

4. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design patterns: Elements of Reusable object-oriented software*, Addison-Wesley, 1995.

5. Simon Bennett, Steve Mc Robb and Ray Farmer, *Object Oriented Systems Analysis and Design Using UML*, Fourth Edition, Mc-Graw Hill Education, 2010.

6. Martin Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Third edition, Addison Wesley, 2003.

WEB REFERENCES:

1. http://vlabs.iitkgp.ernet.in/se/ - Virtual Lab at IIT Kharagpur
2. http://codecourse.sourceforge.net/materials/The-Waterfall-Lifecycle-Model.html
3. http://www.agilemodeling.com/
4. http://www.uml.org/
5. http://www.uml-diagrams.org
6. http://www.tutorialspoint.com/uml
7. http://www.gofpatterns.com/
8. http://www.objectmentor.com/resources/articles/umlfsm.pdf
9. http://www.agilemodeling.com/essays/umlDiagrams.html

Course Code	Course Name	L	T	P	C
CS2252	Database Management Systems Laboratory	0	0	4	2

Category: Professional Core Course

a. Preamble:

This course enables the students to learn the commands for creating and manipulating the databases. This course makes students to construct queries for retrieval of required data from database and understand views, sequences and synonyms concepts of SQL. This course enriches the knowledge in the concepts of the functions, procedures, triggers and exception handling in SQL. This course enables the students to develop GUI based application for storage and retrieval of data.

b. COURSE OUTCOMES

After successful completion of the course, the students will be able to:

CO. No.	Course Outcome	Knowledge Level
CO1	Choose appropriate DDL, DML, DCL and TCL commands for creating and manipulating the databases.	K3
CO2	Construct appropriate nested queries, sub queries and join queries for efficient retrieval of data.	K3
CO3	Organize database using views, sequences, and synonyms..	K3
CO4	Implement functions, procedures, triggers and exceptions using PL/SQL.	K3
CO5	Develop a GUI based environment for storage and retrieval of data for a real time application.	K3

c. Course Syllabus **Total: 60 periods**

LIST OF EXPERIMENTS:

1. **WRITE AND EXECUTE SIMPLE QUERIES USING SQL**
   a. DDL, TCL and DCL commands
   b. DML commands
   c. Aggregate Functions

2. **WRITE AND EXECUTE ADVANCED QUERIES USING SQL**
    a. Nested Queries and Sub queries
    b. SQL Join

3. **WRITE AND EXECUTE VIEWS, SYNONYMS, SEQUENCE**

4. **WRITE AND EXECUTE QUERIES USING PL/SQL**
    a. Simple programs

5. **WRITE AND EXECUTE QUERIES USING ADVANCED CONCEPTS OF PL/SQL**
    a. Cursors and Procedures
    b. Functions
    c. Triggers
    d. Exception Handling

6. **IMPLEMENT DATABASE CONNECTIVITY CONCEPTS**
    a. Design a Front End for a real time application
    b. Connect the database with the application

7. **MINI PROJECT**

LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS:

S. No.	Description of Equipment	Quantity Required
1.	Personal Computers (Intel Core i3, HDD 500 GB, 4 GB RAM)	30
2.	Printer	1
3.	Software: XAMPP with Apache, MySQL & PHP (or) MySQL & JAVA.	Open source

d. Learning Resources

REFERENCES

1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, *Database System Concepts*, 6th edition, Tata McGraw Hill, 2017.
2. Ramez Elmasri, Shamkant B. Navathe, "*Fundamentals of Database Systems*", 6th edition, Pearson Education, 2011.

Course Code	Course Name	L	T	P	C
IT2254	Data Structures Laboratory	0	0	4	2

Category: Professional Core Course

a. Preamble

This course enables the students to implement linear and non-linear data structures, understand the different operations of search trees, implement graph traversal algorithms and get familiarized to sorting and searching algorithms.

b. Course Outcomes

After successful completion of the course, the students will be able to:

Co. No.	Course Outcomes	Knowledge Level
CO1	Implement linear data structures - Array, List, Stack and Queue ADTs for problem solving	K3
CO2	Implement non-linear, hierarchical data structure - Trees for problem solving	K3
CO3	Implement non-linear, non-hierarchical data structure - Graph for problem solving	K3
CO4	Implement various Searching and Sorting Algorithms	K3
CO5	Apply appropriate hash functions in a hash ADT to facilitate collision free data storage and retrieval	K3

c. Course Syllabus **Total : 60 periods**

LIST OF EXPERIMENTS:

1. Array implementation of List ADT
2. Linked implementation of List ADT
    a. Singly Linked List -Merging of Two Lists

b. Doubly Linked List - Palindrome Checking

c. Circular Linked List - Josephus Problem

3. Array implementation of Stack and Queue ADTs
4. Linked list implementation of Stack and Queue ADTs
5. Applications of Stack ADT - Implementation and evaluation of Polish Form
6. Applications of Queue ADT - Implementation of Deque
7. Implementation of Binary Search Trees
8. Implementation of AVL Trees
9. Implementation of Heaps using Priority Queues
10. Graph Representation and Traversal Algorithms
    a. Adjacency Matrix Representation
    b. Adjacency List Representation
11. Applications of Graphs - Single Source Shortest Path
12. Implementation of Searching Algorithms
    a. Linear Search
    b. Binary Search
13. Implementation of Sorting Algorithms
    a. Bubble Sort
    b. Insertion Sort
    c. Selection Sort
    d. Shell Sort
    e. Radix Sort
14. Implementation of Hashing Techniques: Open Hashing
    a. Linear Probing
    b. Quadratic Probing

SOFTWARE SPECIFICATIONS:

1. C Compiler

d. Learning Resources

REFERENCES:

1. Horowitz & Sahni, "*Fundamentals of Data Structures in C*", 2nd edition, Orient Publication, 2008.

2. Aho, Hopcroft & Ullman, "*Data Structures and Algorithms*", Addison Wesley, 1983.

Course Code	Course Name	L	T	P	C
IT2255	Operating Systems Laboratory	0	0	4	2

Category: Professional Core Course

a. Preamble

This course enables the students to practice basic commands of operating systems, execute system calls of UNIX operating system and practice basic shell programming, implement process synchronization mechanisms in operating systems, learn various process management schemes in operating systems, practice with different memory management mechanisms and implement the file allocation techniques.

b. Course Outcomes

After successful completion of the course, the students will be able to:

Co. No.	Course Outcomes	Knowledge Level
CO1	Practice UNIX commands, system calls and write shell scripts involving selection and loops	K3
CO2	Create processes and implements inter process communication with synchronization	K3
CO3	Execute various CPU scheduling algorithms	K3
CO4	Implement deadlock avoidance and detection algorithms	K3
CO5	Illustrate various memory allocation methods, page replacement algorithms, file allocation and organization techniques	K2

c. Course Syllabus **Total : 60 periods**

LIST OF EXPERIMENTS

1. **Unix Commands**
   a. Basics of UNIX commands
2. **System Calls - Commands and Implementation Using C**
   a. Write programs using the following system calls of UNIX operating system fork, exec, getpid, exit, wait, close, stat, opendir, readdir
   b. Write C programs to simulate UNIX commands like cp, ls, grep, etc.
3. **Shell Programming**
   a. Shell Programming

4. CPU Scheduling

a. Write C programs to implement the various CPU Scheduling Algorithms

5. IPC - Semaphores, Shared Memory

a. Implementation of Semaphores

b. Implementation of Shared memory and IPC

6. Deadlock Detection and Avoidance

a. Bankers Algorithm for Deadlock Avoidance

b. Implementation of Deadlock Detection Algorithm

7. Threading and Its Synchronization

a. Implementation of Threading & Synchronization Applications

8. Memory Allocation Methods

a. Implementation of the following Memory Allocation Methods for fixed partition

i) First Fit ii) Worst Fit iii) Best Fit

b. Implementation of Paging Technique of Memory Management

9. Page Replacement Algorithms

a. Implementation of the following Page Replacement Algorithms

i) FIFO ii) LRU iii) LFU

10. File Allocation and Organization Techniques

a. Implementation of the various File Organization Techniques

b. Implementation of the following File Allocation Strategies

i) Sequential ii) Indexed iii) Linked

SOFTWARE SPECIFICATIONS:

1. Linux Operating System
2. CC / GCC Compiler

d. Learning Resources

REFERENCES:

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, "*Operating System Concepts Essentials*", John Wiley & Sons Inc., 10th Edition, 2018
2. Andrew S. Tanenbaum, "*Modern Operating Systems*", Addison Wesley, Second Edition, 2001.