HERITAGE INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Under MAKAUT)

SYLLABUS OF 3rd SEMESTER

Department of Computer Science & Engineering

A. THEORY COURSES

Course Name: Data Structures & Algorithms					
Course Code: CSEN2101					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	4	**0**	**0**	**4**	**4**

1. Course Outcomes

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

CSEN2101.1. Understand and remember the basics of data structures and how time complexity analysis is applicable to different types of algorithms.

CSEN2101.2. Understand the significance and utility of different data structures and the context of their application. (For example, the queue in front of ticket counters uses first-in-first-out paradigm in a linear data structure)

CSEN2101.3. Apply different types of data structures in algorithms and understand how the data structures can be useful in those algorithms.

CSEN2101.4. Analyse the behaviour of different data structures in algorithms. (For example, given an algorithm that uses a particular data structure, how to calculate its space and time complexity.)

CSEN2101.5. Evaluate solutions of a problem with different data structures and thereby understand how to select suitable data structures for a solution. (For example, what are the different ways to find the second largest number from a list of integers and which solution is the best.)

CSEN2101.6. Evaluate different types of solutions (e.g. sorting) to the same problem.

2. Detailed Syllabus

Module 1 [8L]

Introduction: Why do we need data structure? Concepts of data structures: a) Data and data structure b) Abstract Data Type and Data Type; Algorithms and programs, basic idea of pseudo-code. Algorithm efficiency and analysis, time and space analysis of algorithms – Big O, Ω, Θ, notations.

Array: Different representations – row major, column major. Sparse matrix - its implementation and usage. Array representation of polynomials.

Linked List: Singly linked list, circular linked list, doubly linked list, linked list representation of polynomial and applications.

Module 2 [8L]

Stack and Queue: Stack and its implementations (using array, using linked list), applications. Queue, circular queue, deque. Implementation of queue- both linear and circular (using array, using linked list), applications. Implementation of deque- with input and output restriction.

Recursion: Principles of recursion – use of stack, differences between recursion and iteration, tail recursion. Applications - The Tower of Hanoi, Eight Queens Puzzle (Concept of Backtracking).

Module 3 [13L]

Trees: Basic terminologies, forest, tree representation (using array, using linked list). Binary trees - binary tree traversal (pre-, in-, post- order), threaded binary tree (left, right, full) - non-recursive traversal algorithms using threaded binary tree, expression tree. Binary search tree- operations (creation, insertion, deletion, searching). Height balanced binary tree – AVL tree (insertion, deletion with examples only). B- Trees – operations (insertion, deletion with examples only).

Graphs: Graph definitions and Basic concepts (directed/undirected graph, weighted/un-weighted edges, sub-graph, degree, cut vertex/articulation point, complete graph, simple path, simple cycle). Graph representations/storage implementations – adjacency matrix, adjacency list, Graph traversal and connectivity – Depth-first search (DFS), Breadth-first search (BFS) – concepts of edges used in DFS and BFS (tree-edge, back-edge, cross-edge, forward-edge), applications.

Module 4 [11L]

Sorting Algorithms: Bubble sort and its optimizations, Cocktail Shaker Sort, Insertion sort, Selection sort, Quicksort (Average Case Analysis not required), Heap sort (concept of max heap, application – priority queue), Counting Sort, Radix sort.

Searching: Sequential search, Binary search, Interpolation search.

Hashing: Hashing functions, collision resolution techniques (Open and closed hashing).

3. Textbooks

1. Fundamentals of Data Structures of C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-freed.
2. Data Structures in C, Aaron M. Tenenbaum.
3. Data Structures, S. Lipschutz.
4. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.

4. Reference Books

1. Data Structures and Program Design In C, 2/E, Robert L. Kruse, Bruce P. Leung.

Course Name: Discrete Mathematics					
Course Code: CSEN2102					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	4	**0**	**0**	**4**	**4**

1. Course Outcomes

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

CSEN2102.1. Interpret the problems that can be formulated in terms of graphs and trees.
CSEN2102.2. Explain network phenomena by using the concepts of connectivity, independent sets, cliques, matching, graph coloring etc.
CSEN2102.3. Achieve the ability to think and reason abstract mathematical definitions and ideas relating to integers through concepts of well-ordering principle, division algorithm, greatest common divisors and congruence.
CSEN2102.4. Apply counting techniques and the crucial concept of recurrence to comprehend the combinatorial aspects of algorithms.
CSEN2102.5. Analyze the logical fundamentals of basic computational concepts.
CSEN2102.6. Compare the notions of converse, contrapositive, inverse etc. in order to consolidate the comprehension of the logical subtleties involved in computational mathematics.

2. Detailed Syllabus

Module 1 [10L]

Graph Theory: Tree, Binary Tree, Spanning Tree. Walk, Path, Cycle, Hamiltonian Graph, The Travelling Salesman Problem, Euler Graph, The Chinese Postman Problem. Planar Graph, Euler's Formula for Planar Graph and Related Problems. Examples of Non-Planar Graphs. Kuratowski's Theorem. Matching and Augmenting Paths, Hall's Marriage Theorem and Related Problems. Vertex Coloring, Chromatic Polynomials.

Module 2 [10L]

Number Theory: Well Ordering Principle, Principle of Mathematical Induction, Divisibility theory and properties of divisibility, Fundamental Theorem of Arithmetic, Euclidean Algorithm for finding greatest common divisor (GCD) and some basic properties of GCD with simple examples, Congruence, Residue classes of integer modulo n ($\mathbb{Z}_n$) and its examples.

Module 3 [10L]

Combinatorics: Counting Techniques: Permutations and Combinations, Distinguishable and Indistinguishable Objects, Binomial Coefficients, Generation of Permutations and Combinations, Pigeon-hole Principle, Generalized Pigeon-Hole Principle, Principle of Inclusion and Exclusion, Generating Functions and Recurrence Relations: Solving Recurrence Relations Using Generating Functions and other Methods, Divide-and-Conquer Methods, Formulation and Solution of Recurrence Relations in Computer Sorting, Searching and other Application Areas.

Module 4 [12L]

Propositional Calculus: Propositions, Logical Connectives, Truth Tables, Conjunction, Disjunction, Negation, Implication, Converse, Contra positive, Inverse, Biconditional Statements, Logical Equivalence, Tautology, Normal Forms, CNF and DNF, Predicates, Universal and Existential Quantifiers, Bound and Free Variables, Examples of Propositions with Quantifiers.

3. Textbooks

1. Discrete Mathematics and its Applications, Kenneth H. Rosen, Tata McGraw- Hill.
2. Discrete Mathematics, T Veerarajan, Tata McGraw- Hill.

4. Reference Books

1. Elements of Discrete Mathematics: A Computer Oriented Approach, C L Liu and D P Mohapatra, McGraw Hill.
2. Discrete Mathematical Structure and Its Application to Computer Science, J.P. Tremblay and R. Manohar, McGraw Hill.
3. Discrete Mathematics for Computer Scientists and Mathematicians, J.L.Mott, A. Kandel and T.P.Baker, Prentice Hall
4. Discrete Mathematics,Norman L. Biggs, Seymour Lipschutz, Marc Lipson, Oxford University Press, Schaum's Outlines Series.
5. Higher Algebra (Classical), S.K. Mapa, Sarat Book Distributors.
6. Introduction to Graph Theory (2nd Ed), D G West, Prentice-Hall of India, 2006.

Course Name: Analog Circuits					
Course Code: ECEN2101					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	3	**0**	**0**	**3**	**3**

1. Course Outcomes

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

ECEN2101.1. Apply the previous knowledge gathered from Basic Electrical and Basic Electronics papers.
ECEN2101.2. Understand the concepts of BJT, MOSFET and biasing techniques of BJT and MOSFET based amplifier circuits.
ECEN2101.3. Analyse frequency response of amplifier circuits.
ECEN2101.4. Design different types sinusoidal oscillators and multi-vibrator circuits.
ECEN2101.5. Construct algebraic equations-based amplifier and analog computers using OP-AMP
ECEN2101.6. Design stable high-gain amplifier circuits.

2. Detailed Syllabus

Module 1 [9L]

Basic concepts and device biasing: Analog, discrete and digital signals. Diode: piecewise-linear model, clipping and clamping operation. BJT biasing circuits, Q-point and stability.
Small Signal analysis of Amplifiers: Small signal (h-parameter and r_e model) analysis of BJT CE mode amplifier circuit (derive input impedance, output impedance, voltage gain, current gain for the amplifiers).

Module 2 [9L]

Frequency Responses of Amplifiers: Frequency response of CE mode RC-coupled amplifier; effect of external and parasitic capacitors on cut-off frequencies.
Feedback & Oscillator Circuits: Concept of feedback, Effects of negative feedback in amplifiers, Oscillators circuits: Phase-shift, Wien-Bridge, Hartley, Colpitts and crystal Oscillators.

Module 3 [7L]

Fundamentals of OPAMP: Basic building blocks of OPAMP: Differential Amplifiers, Current source and current mirror circuits. Types of differential amplifiers, AC and DC analysis of differential amplifiers; Characteristics of an ideal OPAMP.
Applications of OPAMP: Inverting and non-inverting OPAMP amplifiers, Log-antilog amplifiers, Instrumentation amplifier, Precision rectifiers, basic comparator, Schmitt Trigger.

Module 4 [7L]

Power Amplifiers: Concepts and operations of Class A, B and AB amplifiers; Calculation of DC power, AC power and efficiency of these amplifiers.
Applications Analog IC: Description of 555 Timer IC, astable and mono-stable operations using 555. Study of 78XX and 79XX voltage regulator ICs.

3. Textbooks

1. Microelectronic Circuits by Adel S. Sedra, Kenneth C. Smith.
2. Electronics Devices and Circuits by Robert L. Boylestad, Louis Nashelskey.
3. Fundamentals of Microelectronics by Behzad Razavi.
4. Integrated electronics by Jacob Millman, Christos C. Halkias.

Course Name: Digital Logic					
Course Code: ECEN2104					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	3	**0**	**0**	**3**	**3**

1. Course Outcomes

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

ECEN2104.1. Students will learn Binary Number system, and logic design using combinational gates.
ECEN2104.2. Students will design applications of Sequential Circuits.
ECEN2104.3. Students will design Finite State Machines.
ECEN2104.4. Students will learn Memory classifications.
ECEN2104.5. Students will learn basics of CMOS logic.
ECEN2104.6. Students will be prepared to learn various digital component design as used in VLSI applications.

2. Detailed Syllabus

Module 1 [10L]

Binary System, Boolean Algebra and Logic Gates: Data and number systems; Binary, Octal and Hexadecimal representation and their conversions, BCD, Gray codes, excess 3 codes and their conversions; Signed binary number representation with 1's and 2's complement methods, Binary arithmetic. Boolean algebra, De-Morgan's theorem, Various Logic gates- their truth tables and circuits, universal logic gates, Representation in SOP and POS forms; Minimization of logic expressions by algebraic method, Karnaugh-map method, Quine-McCluskey method.

Module 2 [10L]

Arithmetic Circuits: Adder circuit – Ripple Carry Adder, CLA Adder, CSA, and BCD adder, subtractor circuit.
Combinational Circuit: Encoder, Decoder, Comparator, Multiplexer, De-Multiplexer and parity Generator. Shannon's Expansion Theorem, Realization of logic functions using Mux, Parity Generators.

Module 3 [10L]

Sequential Logic: Basic memory elements, S-R, J-K, D and T Flip Flops, Sequential circuits design methodology: State table and state diagram, State Reduction Method, Circuit Excitation and Output tables, Derivation of Boolean functions; Finite State Machine Design using Sequential circuit design methodology, various types of Registers (with Parallel load, shift Registers) and Counters (asynchronous ripple counters, synchronous counters: binary, BCD, Johnson).

Module 4 [6L]

Memory Systems: Concepts and basic designs of RAM (SRAM & DRAM), ROM, EPROM, EEPROM, Programmable logic devices and gate arrays (PLAs and PLDs)
Logic families: NMOS and CMOS, their operation and specifications. Realization of basic gates using above logic families, Open collector & Tristate gates, wired-AND and bus operations.

3. Textbooks

1. Digital Logic and Computer Design, Morris M. Mano, PHI.
2. Digital Principles & Applications, 5th Edition, Leach & Malvino, Mc Graw Hill Company.
3. Modern Digital Electronics, 2nd Edition, R.P. Jain. Tata Mc Graw Hill Company Limited.
4. Digital Logic Design, Fourth Edition - Brian Holdsworth & Clive Woods.
5. Digital Integrated Electronics, H.Taub & D.Shilling, Mc Graw Hill Company Limited.

4. Reference Books

1. Digital Design: Principles and Practices: John F. Wakerly.
2. Fundamental of Digital Circuits, A. Anand Kumar, PHI.

Course Name: Human Values and Professional Ethics					
Course Code: HMTS2001					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	3	**0**	**0**	**3**	**3**

1. Course Outcomes

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

HMTS2001.1. Be aware of the value system and the importance of following such values at workplace.
HMTS2001.2. Learn to apply ethical theories in the decision-making process.
HMTS2001.3. Follow the ethical code of conduct as formulated by institutions and organizations.
HMTS2001.4. Implement the principles governing work ethics.
HMTS2001.5. Develop strategies to implement the principles of sustainable model of development.
HMTS2001.6. Implement ecological ethics wherever relevant and also develop eco-friendly technology.

2. Detailed Syllabus

Module 1 [10L]

Human society and the Value System: Values: Definition, Importance and application, Formation of Values: The process of Socialization, Self and the integrated personality, Morality, courage, integrity.
Types of Values: Social Values: Justice, Rule of Law, Democracy, Indian Constitution, Secularism; Aesthetic Values: Perception and appreciation of beauty; Organizational Values: Employee: Employer--- rights, relationships, obligations; Psychological Values: Integrated personality and mental health; Spiritual Values and their role in our everyday life; Value Spectrum for a Good Life, meaning of Good Life.
Value Crisis in Contemporary Society: Value crisis at: Individual Level, Societal Level, Cultural Level; Value Crisis management: Strategies and Case Studies.

Module 2 [10L]

Ethics and Ethical Values, Principles and theories of ethics, Consequential and non-consequential ethics, Egotism, Utilitarianism, Kant's theory and other non-consequential perspectives, Ethics of care, justice and fairness, rights and duties.

Ethics: Standardization, Codification, Acceptance, Application.

Types of Ethics: Ethics of rights and Duties, Ethics of Responsibility, Ethics and Moral judgment, Ethics of care
Ethics of justice and fairness, Work ethics and quality of life at work.

Professional Ethics: Ethics in Engineering Profession; moral issues and dilemmas, moral autonomy (types of inquiry), Kohlberg's theory, Gilligan's theory (consensus and controversy), Code of Professional Ethics Sample Code of ethics like ASME, ASCE. IEEE, Institute of Engineers, Indian Institute of materials management, Institute of Electronics and telecommunication engineers, Violation of Code of Ethics---conflict, causes and consequences

Engineering as social experimentation, engineers as responsible experimenters (computer ethics, weapons development), Engineers as managers, consulting engineers, engineers as experts, witnesses and advisors, moral leadership, Conflict between business demands and professional ideals, social and ethical responsibilities of technologies.

Whistle Blowing: Facts, contexts, justifications and case studies

Ethics and Industrial Law: Institutionalizing Ethics: Relevance, Application, Digression and Consequences.

Module 3 [10L]

Science, Technology and Engineering: Science, Technology and Engineering as knowledge and profession: Definition, Nature, Social Function and Practical application of science; Rapid Industrial Growth and its Consequences; Renewable and Non- renewable Resources: Definition and varieties; Energy Crisis; Industry and Industrialization; Man and Machine interaction; Impact of assembly line and automation; Technology assessment and Impact analysis; Industrial hazards and safety; Safety regulations and safety engineering; Safety responsibilities and rights; Safety and risk, risk benefit analysis and reducing risk; Technology Transfer: Definition and Types; The Indian Context.

Module 4 [6L]

Environment and Eco- friendly Technology: Human Development and Environment, Ecological Ethics/Environment ethics Depletion of Natural Resources: Environmental degradation, Pollution and Pollution Control, Eco-friendly Technology: Implementation, impact and assessment, Sustainable Development: Definition and Concept, Strategies for sustainable development, Sustainable Development: The Modern Trends, Appropriate technology movement by Schumacher and later development, Reports of Club of Rome.

3. Reference Books

1. Human Values, Tripathi, A.N., New Age International, New Delhi, 2006.
2. Classical Sociological Theory, Ritzer, G., The McGraw Hill Companies, New York,1996.
3. Postmodern Perspectives on Indian Society, Doshi, S.L., Rawat Publications, New Delhi,2008.
4. Sustainable Development, Bhatnagar, D.K., Cyber Tech Publications, New Delhi, 2008.
5. The age of Spiritual Machines, Kurzwell, R., Penguin Books, New Delhi,1999.
6. Social Problems in Modern Urban Society, Weinberg, S.K., Prentice Hall,Inc.,USA, 1970.
7. Sociology, Giddens, Anthony 2009, London: Polity Press (reprint 13th Edition).

B. LABORATORY COURSES

Course Name: Data Structure & Algorithms Lab					
Course Code: CSEN2151					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	0	**0**	**3**	**3**	**1.5**

1. Course Outcomes

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

CSEN2151.1. To understand linear and non-linear data structures.
CSEN2151.2. To understand different types of sorting and searching techniques.
CSEN2151.3. To know how to create an application specific data structure.
CSEN2151.4. To solve the faults / errors that may appear due to wrong choice of data structure.
CSEN2151.5. To analyse reliability of different data structures in solving different problems.
CSEN2151.6. To evaluate efficiency in terms of time and space complexity, when different data structures are used to solve same problem.

2. Detailed Syllabus

Day 1: Time and Space Complexity

Lab Assignment

Create three different 10; 000 10; 000 matrices matrixOne, matrixTwo and result-Matrix, using dynamic memory allocation. Initialize matrixOne and matrixTwo by using rand() or srand() function, limit the values from 0 to 9. Multiply matrixOne and matrixTwo into resultMatrix.

While execution, open another terminal and use top command to see the usage of memory by the process. Calculate the time taken for the execution of the program.
Repeat the same exercise for 100,000 x 100,000matrices.

Home Assignment

Write a program (WAP) to check whether a matrix is i) identity, ii) diagonal. WAP to reverse the elements of an array without using any other variable.

Day 2: Array

Lab Assignment

WAP to add two polynomials using array. Minimize the memory usage as much as you can.
WAP to convert a matrix into its sparse representation (triple format). Once represented in sparse format, do not revert back to the matrix format any-more. Manipulate the sparse representation to find the transpose of the matrix (which should also be in sparse representation).
Calculate and find out whether using triple format for your example is advantageous or not.

Home Assignment

WAP to multiply two polynomials. Minimize usage of memory.
WAP to add two matrices using sparse representation. Manipulation of data should be done in sparse format.

Day 3: Singly Linked List

Lab Assignment

Write a menu driven program to implement a singly linked list with the operations:
i) create the list ii) insert any element in any given position (front, end or intermediate)
iii) delete an element from any given position (front, end or intermediate) iv)display the list

Home Assignment

Write a menu driven program to implement a singly linked list with the operations:
i) count the number of nodes ii) reverse the list

Day 4: Circular and Doubly Linked List

Lab Assignment

Write a menu driven program to implement a circular linked list with the operations:
i) create the list ii) insert any element in any given position (front, end or intermediate)
iii) delete an element from any given position (front, end or intermediate) iv)display the list

Home Assignment

Write a menu driven program to implement a doubly linked list with the operations:
i) create the list ii) insert any element in any given position (front, end or intermediate)
iii) delete an element from any given position (front, end or intermediate) iv)display the list

Day 5: Stack, Queue - with array

Lab Assignment

Write a menu driven program to implement stack, using array, with
i) push, ii) pop, iii) display, iv) exit operations.
WAP to evaluate a postfix expression.
Write a menu driven program to implement a queue, using array, with
i) insert, ii) delete, iii) display, iv) exit operations

Home Assignment

WAP to convert an infix expression to its corresponding postfix operation.
Write a menu driven program to implement a double-ended queue, using array, with the following operations:
i) insert (from front, from rear) ii) delete (from front, from rear)
iii) display iv) exit operations

Day 6: Stack, Queue - with linked list

Lab Assignment

Write a menu driven program to implement a stack, using linked list, with
i) push, ii) pop, iii) exit operations

Home Assignment

Write a menu driven program to implement a queue, using linked list, with
i) insert, ii) delete, iii) exit operations

Day 7: Circular Queue, Deque - with linked list
Lab Assignment
Write a menu driven program to implement a circular queue using linked list with
i) insert, ii) delete, iii) exit operations
Home Assignment
Write a menu driven program to implement a double-ended queue, using linked list, with the following operations:
i) insert (from front, rear), ii) delete (from front, rear), iii) exit operations

Day 8: Binary Search Tree (BST)
Lab Assignment
Write a program, which creates a binary search tree (BST). Also write the functions to insert, delete (all possible cases) and search elements from a BST.
Home Assignment
Write three functions to traverse a given BST in the following orders:
i) in-order, ii) pre-order, iii) post-order.
Display the elements while traversing.

Day 9: Searching
Lab Assignment
WAP to implement,
i) Linear Search, ii) Binary Search (iterative)
NB: As a pre-processing step, use bubble-sort to sort the elements in the search space.
WAP to generate integers from 1 to n (input parameter) in random order and guarantees that no number appears twice in the list. While the number sequence is being generated, store it in a text file.
Home Assignment
WAP to implement binary search recursively.

Day 10: Sorting
Lab Assignment
Write different functions for implementing,
i) Bubble sort, ii) Cocktail shaker sort, iii) Quick Sort.
Plot a graph of n vs. time taken, for n= 100, 1000, 10,000 and 100,000 to com-pare the performances of the sorting methods mentioned above. Use the second assignment of Day 9 to generate the data, using the given n values.
Home Assignment
Write different functions for implementing,
i) Insertion sort, ii) Merge sort.

Day 11: Graph Algorithms
Lab Assignment
Read a graph (consider it to be undirected) from an edge-list and store it in an adjacency list.
Use the adjacency list to run DFS algorithm on the graph and print the node labels. Detect and count the back-edges.
Home Assignment
WAP to implement BFS algorithm of a given graph (similarly as described for DFS, instead of back-edges count cross-edges).

3. Textbooks

1. Fundamentals of Data Structures of C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-freed.
2. Data Structures in C, Aaron M. Tenenbaum.
3. Data Structures, S. Lipschutz.
4. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.

4. Reference Books

1. Data Structures and Program Design In C, 2/E, Robert L. Kruse, Bruce P. Leung.

Course Name: Software Tools Lab					
Course Code: CSEN2152					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	0	**0**	**3**	**3**	**1.5**

1. Course Outcomes

After completion of the course, students will be able to:
CSEN2152.1. Learn the concept and use of an integrated development environment.

CSEN2152.2. Identify different compilation options in gcc and develop static and shared libraries.
CSEN2152.3. Analyze the errors in a code using gdb and valgrind.
CSEN2152.4. Analyse a code with code coverage testing and know how to speed up execution using profiling tools.
CSEN2152.5. Compose a makefile and use the make utility to automate compilations.
CSEN2152.6. Understand the need for version control and learn effective methods to do the same.

2. Detailed Syllabus

1. **CodeLite IDE [Code::Blocks]:** Learn to use CodeLite IDE for writing C/C++ programming languages.
2. **Compiling with gcc:** Learn all the command line options for compiling C programs in the Unix environment using gcc.
3. **Static and Dynamic Library:** Understand the linking phase of a C program by creating and using static and dynamic libraries.
4. **Debugging with gdb:** gdb is the standard C/C++ debugger to debug your code. Learn to interact with gdb directly via a shell, or use a graphical interface provided by CodeLite IDE.
5. **Memory profiling with valgrind:** Learn to use valgrind which is a critical tool for helping one to find memory leaks in the program: malloc without free, accessing an array outside its bounds, etc.
6. **Code coverage testing with gcov:** Learn about good testing using gcov to make sure the tests are exercising all the branches in the code.
7. **Runtime profiling with gprof:** Learn about using gprof which is a very useful profiling tool for speeding up execution speed of a program: it will show where your program is spending most of its time, so one can know about the most important code to optimize
8. **Makefile:** Learn how to use makefile on Unix to properly build an executable.
9. **Git for sharing files and version control:** Learn to setup a repository so that it can sync your local with that on the server. Learn to use CVS for version controlling.

3. Textbooks

1. The Definitive Guide to GCC, William von Hagen, 2nd Edition, 2006, Apress.
2. Linux Debugging and Performance Tuning: Tips and Techniques, Steve Best, Pearson Education,1st Edition, 2006.

4. Reference Books

1. Version control with Git, Jon Loeliger,1st Edition,2009, O'Reilly.
2. The Art of Debugging with GDB, DDD, and Eclipse, Norman Matloff, Peter Jay Salzman, 2008.

Course Name: Digital Logic Lab					
Course Code: ECEN2154					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	0	**0**	**2**	**2**	**1**

1. Course Outcomes

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

ECEN2154.1. Use the concept of Boolean algebra to minimize logic expressions by the algebraic method, K-map method etc.
ECEN2154.2. Construct different Combinational circuits like Adder, Subtractor, Multiplexer, De-Multiplexer, Decoder, Encoder, etc.
ECEN2154.3. Design various types of Registers and Counters Circuits using Flip-Flops (Synchronous, Asynchronous, Irregular, Cascaded, Ring, Johnson).
ECEN2154.4. Realize different logic circuits using ICs built with various logic families.

2. Detailed Syllabus

Choose any ten experiments out of the twelve suggested next:

1. Realization of basic gates using Universal logic gates.
2. Four-bit parity generator and comparator circuits.
3. Code conversion circuits BCD to Excess-3 & vice-versa.
4. Construction of simple 3-to-8 Decoder circuit by 2-to-4 Decoders using logic gates.
5. Design a 4-to-1 Multiplexer using logic gates and use it as a Universal logic module.
6. Realization of SR (Set Reset), JK, and D flip-flops using Universal logic gates.
7. Construction of simple arithmetic logic circuits-Adder, Subtractor.
8. Realization of Asynchronous Up/Down Counter (Count up to 7) using logic gates.
9. Realization of Synchronous Up/Down Counter (Count up to 7) using logic gates.
10. Realization of Shift Registers using logic gates (Serial in Serial out and Parallel in Serial out).
11. Construction of Serial adder circuit using a D Flip-Flop and a Full adder.
12. Design a combinational circuit for BCD to Decimal conversion to drive 7-Segment display using logic gates.

C. HONORS COURSES

Course Name: Probability and Statistical Methods					
Course Code: MATH2111					
Contact Hours per week:	**L**	**T**	**P**	**Total**	**Credit points**
	4	**0**	**0**	**4**	**4**

1. Course Outcomes

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

MATH2111.1. Articulate the axioms (laws) of probability.

MATH2111.2. Compare and contrast different interpretations of probability theory and take a stance on which might be preferred.

MATH2111.3. Formulate predictive models to tackle situations where deterministic algorithms are intractable.

MATH2111.4. Summarize data visually and numerically.

MATH2111.5. Assess data-based models.

MATH2111.6. Apply tools of formal inference.

2. Detailed Syllabus

Module 1 [10L]

Probability-I (Single variable probability distributions): Review of basic probability: Axiomatic definition, Addition and Multiplication law, Conditional probability and Bayes' Theorem, Expectation and Variance of single variable discrete and continuous distributions, Normal approximation to Binomial and Poisson Distribution, Exponential and Multinomial distribution, Moment generating and characteristic functions, Limit theorems: Markov's inequality and Chebyshev's inequality with examples.

Module 2 [10L]

Probability-II (Joint Distribution and Markov Chains): Joint distribution using joint probability mass/density function, Finding marginal pmf/pdf from joint distribution, Multiplicative property of joint pmf/pdf in case of independent random variables, Markov Chains: Introduction, Chapman-Kolmogorov equations, Classification of states, Some applications: Gambler's Ruin Problem.

Module 3 [10L]

Statistics-I: Moments, Skewness and Kurtosis, Binomial, Poisson and Normal - evaluation of statistical parameters for these three distributions, Covariance, Correlation and Regression, Spearman's Rank Correlation coefficient, Curve fitting: Straight line and parabolas.

Module 4 [10L]

Statistics-II: Population and Samples, The sampling distribution of mean (standard deviation known), The sampling distribution of mean (standard deviation unknown), Point and Interval estimation, Tests of Hypotheses, Null Hypotheses and Tests of Hypotheses with examples.

3. Textbooks

1. Probability and Statistics for Engineers, Richard A Johnson, Pearson Education.
2. Groundwork of Mathematical Probability and Statistics, Amritava Gupta, Academic Publishers.

4. Reference Books

1. Introduction to Probability Models, S.M. Ross, Elsevier.
2. Fundamentals of Mathematical Statistics, S.C. Gupta and V.K. Kapoor, Sultan Chand and Sons.
3. An Introduction to Probability theory and its applications Vol-I, W. Feller, John Wiley and Sons.

Course Title : Introduction to Analog Circuits					
Course Code : ECEN2105					
Contact Hours per week	L	T	P	Total	Credit Points
	3	0	0	3	3

After going through this course, the students will be able to

1) Classify different semiconductor materials based on their conductivity.
2) Understand P-N Junction characteristics and circuit applications.
3) Understand Bipolar Junction Transistor characteristics and applications.
4) Categorize different Field Effect Transistors based on their structure and working principle
5) Understand feedback and its effects.
6) Apply basic concepts of feedback and design operational amplifier based circuits.

Module I [10 L]

Introduction to semiconductor materials:
Crystalline materials, energy band theory, Conductors, Semiconductors and Insulators, Concept of Fermi energy level, intrinsic and extrinsic semiconductors.

Diodes and Diode Circuits:
Formation of p-n junction, energy band diagram, forward and reverse biased configurations, V-I characteristics, DC load line, breakdown mechanisms - Zener and Avalanche breakdown, Rectifier circuits: half wave and full wave rectifiers.

Module II [8 L]

Bipolar Junction Transistors (BJT) and BJT Circuits:
PNP and NPN BJT structures, different operating modes of BJT, input & output V-I characteristics of CE and CB configurations. Concept of Biasing: DC load line, Q-point, basic amplifier circuits using BJT.

Module III [8 L]

Field Effect Transistors (FET) and FET Circuits:
Classification of FET, basic structure and operation of Junction Field Effect Transistor (n-channel),V-I characteristics.

Metal Oxide Semiconductor Field Effect Transistor (MOSFET): Enhancement and depletion type MOSFETs, drain and transfer characteristics, Basic amplifier circuits using FET.

Module IV [8 L]

Operational Amplifier:

OP-AMP characteristics, Basic Comparator, Inverting and Non-inverting amplifiers, Adder, Subtractor, integrator and differentiator, Logarithmic amplifier and Anti Logarithmic amplifier, Multiplier, Schmitt trigger circuit, Analog computer design using OP-AMP.