Syllabus

for

B.TECH. COMPUTER SCIENCE AND ENGINEERING

of

Second Year

(Effective from the Session: 2014-15)
### B.TECH COMPUTER SCIENCE AND ENGINEERING
**STUDY & EVALUATION SCHEME**

#### 2nd Year

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course Code</th>
<th>Subject</th>
<th>Periods</th>
<th>Evaluation Scheme</th>
<th>Subject Total</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L  T  P</td>
<td>Sessional Exam</td>
<td>ESE</td>
<td>Total</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CT  TA</td>
<td>Total</td>
<td></td>
</tr>
</tbody>
</table>

**THEORY SUBJECT**

1. NAS-301/NOE-031 to NOE-038 Mathematics III/Applied Engineering Chemistry 3 1 0 30 20 50 100 150 4
2. NEC 309 Digital Logic Design 3 1 0 30 20 50 100 150 4
3. NCS 301 Data Structures Using C 3 1 0 30 20 50 100 150 4
4. NCS 302 Discrete Structures And Graph Theory 3 1 0 30 20 50 100 150 4
5. NHU301/NHU302 Industrial Psychology/Industrial Sociology 2 0 0 15 10 25 50 75 2
6. NCS 303 Computer Based Numerical And Statistical Techniques 2 1 0 15 10 25 50 75 3
7. AUC-001/AUC-002 Human Values & Professional Ethics/Cyber Security 2 0 0 15 10 25 50 75*

**PRACTICAL/DESIGN/DRAWING**

7. NEC 359 Digital Logic Design Lab 0 0 3 10 10 20 30 50 1
8. NCS 351 Data Structures Using C Lab 0 0 3 10 10 20 30 50 1
9. NCS 353 Numerical Techniques Lab 0 0 2 10 10 20 30 50 1
10. NCS 355 Advance Programming Lab 0 0 2 10 10 20 30 50 1
11. NGP 301 GP 50 50

TOTAL 18 5 10 1000 25

---

**B.TECH COMPUTER SCIENCE AND ENGINEERING**

**STUDY & EVALUATION SCHEME**

#### 2nd Year

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course Code</th>
<th>Subject</th>
<th>Periods</th>
<th>Evaluation Scheme</th>
<th>Subject Total</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L  T  P</td>
<td>Sessional Exam</td>
<td>ESE</td>
<td>Total</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CT  TA</td>
<td>Total</td>
<td></td>
</tr>
</tbody>
</table>

**THEORY SUBJECT**

1. NOE-041 to NOE-048/NAS-401 Mathematics III/Applied Engineering Chemistry 3 1 0 30 20 50 100 150 4
2. NHU401/NHU402 Introduction to Microprocessor 3 1 0 30 20 50 100 150 4
3. NEC 409 Operating System 3 1 0 30 20 50 100 150 4
4. NCS 401 Theory Of Automata and Formal Language 3 1 0 30 20 50 100 150 4
5. NCS 402 Computer Graphics 2 1 0 15 10 25 50 75 3
6. AUC-002/AUC-001 Cyber Security /Human Values & Professional Ethics 2 0 0 15 10 25 50 75*

**PRACTICAL/DESIGN/DRAWING**

7. NEC 455 Microprocessor Lab 0 0 3 10 10 20 30 50 1
8. NCS 451 Operating System Lab 0 0 3 10 10 20 30 50 1
9. NCS 453 Computer Graphics Lab 0 0 2 10 10 20 30 50 1
10. NCS 455 Functional and Logic Programming Lab 0 0 2 10 10 20 30 50 1
11. NGP 401 GP 50 50

TOTAL 18 5 10 1000 25

*Human values & Professional Ethics/Cyber Security will be offered as a compulsory audit course for which passing marks are 30% in End Semester Examination and 40% in aggregate.

The details of Science Based Electives are to be provided by The Boards of Studies of Science Subjects; these are common to all branches.
NEC-309: DIGITAL LOGIC DESIGN

Unit-I
Digital Design and Binary Numbers:
Binary Arithmetic, Negative Numbers and their Arithmetic, Floating point representation, Binary Codes, Cyclic Codes, Error Detecting and Correcting Codes, Hamming Codes.
Minterm and Maxterm Realization of Boolean Functions, Gate-level minimization: The map method up to four variable, don’t care conditions, SOP and POS simplification, NAND and NOR implementation, Quine Mccluskey Method (Tabular method).

Unit-II
Combinational Logic:
Combinational Circuits, Analysis Procedure, Design Procedure, Binary Adder-Subtractor, Code Converters, Parity Generators and Checkers, Decimal Adder, Binary Multiplier, Magnitude Comparator, Decoders, Encoders, Multiplexers, Hazards and Threshold Logic

Unit-III
Memory and Programmable Logic Devices:
Semiconductor Memories, RAM, ROM, PLA, PAL, Memory System design.

Unit-IV
Synchronous Sequential Logic:
Sequential Circuits, Storage Elements: Latches, Flip Flops, Analysis of Clocked Sequential circuits, state reduction and assignments, design procedure.
Registers and Counters: Shift Registers, Ripple Counter, Synchronous Counter, Other Counters.

Unit-V
Asynchronous Sequential Logic: Analysis procedure, circuit with latches, design procedure, reduction of state and flow table, race free state assignment, hazards.

References:

NCS-301: DATA STRUCTURES USING – C

Unit - I
Abstract Data Types (ADT)

Unit – II

Unit – III

Unit – IV

Unit – V
Searching: Sequential search, Binary Search, Comparison and Analysis
Internal Sorting: Insertion Sort, Bubble Sort, Quick Sort, Two Way Merge Sort, Heap Sort, Radix Sort, Practical consideration for
Internal Sorting.
Search Trees: Binary Search Trees (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, AVL trees, Introduction to m-way Search Trees, B Trees & B+ Trees.
Hashing: Hash Function, Collision Resolution Strategies
Storage Management: Garbage Collection and Compaction.

References:

NCS-302: DISCRETE STRUCTURES AND GRAPH THEORY

Unit-I
Set Theory: Introduction, Combination of sets, Multisets, Ordered pairs, Set Identities.
Relations: Definition, Operations on relations, Properties of relations, Composite Relations, Equality of relations, Order of relations.
Functions: Definition, Classification of functions, Operations on functions, Recursively defined functions.
Natural Numbers: Introduction, Mathematical Induction, Variants of Induction, Induction with Nonzero Base cases.

Unit-II
Algebraic Structures: Definition, Groups, Subgroups and order, Cyclic Groups, Cosets, Lagrange’s theorem, Normal Subgroups, Permutation and Symmetric groups, Group Homomorphisms, Definition and elementary properties of Rings and Fields, Integers Modulo n.

Unit-III
Partial order sets: Definition, Partial order sets, Combination of partial order sets, Hasse diagram.
Lattices: Definition, Properties of lattices – Bounded, Complemented, Modular and Complete Lattice, Morphisms of lattices.
Boolean Algebra: Introduction, Axioms and Theorems of Boolean algebra, Algebraic manipulation of Boolean expressions, Simplification of Boolean Functions, Karnaugh maps, Logic gates, Digital circuits and Boolean algebra. Combinational and sequential Circuits

Unit-IV
Propositional Logic: Proposition, well formed formula, Truth tables, Tautology, Satisfiability, Contradiction, Algebra of proposition, Theory of Inference, Natural Deduction.
Predicate Logic: First order predicate, well formed formula of predicate, quantifiers, Inference theory of predicate logic.

Unit-V
Trees: Definition, Binary tree, Binary tree traversal, Binary search tree.
Graphs: Definition and terminology, Representation of graphs, Multigraphs, Bipartite graphs, Planar graphs, Isomorphism and Homeomorphism of graphs, Euler and Hamiltonian paths, Graph coloring.
Recurrence Relation & Generating function: Recursive definition of functions, Recursive algorithms, Method of solving recurrences.
Combinatorics: Introduction, Counting Techniques, Pigeonhole Principle

References:
4. R.P. Grimaldi, Discrete and Combinatorial Mathematics, Addison Wesley,
NCS-303: COMPUTER BASED NUMERICAL AND STATISTICAL TECHNIQUES (Half Unit)

UNIT - 1
Introduction to Graph Theory:
Solution of Algebraic and Transcendental Equation:
Bisection Method, Iteration method, Method of false position, Newton-Raphson method, Muller's method, Rate of convergence of Iterative methods

UNIT - 2
Interpolation: Finite Differences, Difference tables Polynomial Interpolation: Newton's forward and backward formula Central difference Formulea: Gauss forward and backward formula, Stirling's, Bessel's, Everett's formula. Interpolation with unequal intervals: Langrange's interpolation, NewtonDivided difference formula.

UNIT - 3

UNIT - 4

UNIT - 5
Statistical Computation: Frequency chart, Curve fitting by method of least squares, fitting of straight lines, polynomials, exponential curves etc, Data fitting with Cubic splines, Regression Analysis, Linear and Non linear Regression, Multiple regression, Statistical Quality Control methods

References:
1. Rajaraman V, "Computer Oriented Numerical Methods", Pearson Education

NEC-359: LOGIC DESIGN LAB

Objective: To understand the digital logic and create various systems by using these logics.
1. Introduction to digital electronics lab- nomenclature of digital ICs, specifications, study of the data sheet, concept of Vcc and ground, verification of the truth tables of logic gates using TTL ICs.
2. Implementation of the given Boolean function using logic gates in both SOP and POS forms.
3. Verification of state tables of RS, JK, T and D flip-flops using NAND & NOR gates.
4. Implementation and verification of Decoder/De-multiplexer and Encoder using logic gates.
5. Implementation of 4x1 multiplexer using logic gates.
6. Implementation of 4-bit parallel adder using 7483 IC.
7. Design, and verify the 4-bit synchronous counter.
8. Design, and verify the 4-bit asynchronous counter.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.
NCS-351: DATA STRUCTURE USING C LAB

Program in C or C++ for following:
1. To implement addition and multiplication of two 2D arrays.
2. To transpose a 2D array.
3. To implement stack using array.
4. To implement queue using array.
5. To implement circular queue using array.
6. To implement stack using linked list.
7. To implement queue using linked list.
8. To implement circular queue using linked list.
9. To implement binary tree using linked list.
10. To implement binary search tree using linked list.
11. To implement tree traversals using linked list.
12. To implement BFS using linked list.
13. To implement DFS using linked list.
14. To implement Linear Search.
15. To implement Binary Search.
16. To implement Bubble Sorting.
17. To implement Selection Sorting.
18. To implement Insertion Sorting.
19. To implement Merge Sorting.
20. To implement Heap Sorting.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

NCS-353: NUMERICAL TECHNIQUES LAB

Write Programs in ‘C’ Language:
1. To deduce error involved in polynomial equation.
2. To Find out the root of the Algebraic and Transcendental equations using Bisection, Regula-falsi, Newton Raphson and Iterative Methods. Also give the rate of convergence of roots in tabular form for each of these methods.
3. To implement Newton’s Forward and Backward Interpolation formula.
4. To implement Gauss Forward and Backward, Bessel’s, Sterling’s and Evertt’s Interpolation formula.
5. To implement Newton’s Divided Difference and Langranges Interpolation formula.
6. To implement Numerical Differentiations.
7. To implement Numerical Integration using Trapezoidal, Simpson 1/3 and 0Simpson 3/8 rule.
8. To implement Least Square Method for curve fitting.
9. To draw frequency chart like histogram, frequency curve and pie-chart etc.
10. To estimate regression equation from sampled data and evaluate values of standard deviation, t-statistics, regression coefficient, value of $R^2$ for atleast two independent variables.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

NCS-355: ADVANCE PROGRAMMING LAB

LIST OF EXPERIMENTS:
1. Programs using Functions and Pointers in C
2. Programs using Files in C
3. Programs using Classes and Objects
4. Programs using Operator Overloading
5. Programs using Inheritance, Polymorphism and its types
6. Programs using Arrays and Pointers
7. Programs using Dynamic memory allocation
8. Programs using Templates and Exceptions
9. Programs using Sequential and Random access files
Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

**NEC 409: INTRODUCTION TO MICROPROCESSOR**

**UNIT I**
Introduction to Microprocessor, Microprocessor architecture and its operations, Memory, Input & output devices, Logic devices for interfacing, The 8085 MPU, Example of an 8085 based computer, Memory interfacing.

**UNIT II**
Basic interfacing concepts, Interfacing output displays, Interfacing input devices, Memory mapped I/O, Flow chart symbols, Data Transfer operations, Arithmetic operations, Logic Operations, Branch operation, Writing assembly language programs, Programming techniques: looping, counting and indexing.

**UNIT III**
Additional data transfer and 16 bit arithmetic instruction, Arithmetic operations related to memory, Logic operation: rotate, compare, counter and time delays, Illustrative program: Hexadecimal counter, zero-to-nine, (module ten) counter, generating pulse waveforms, debugging counter and time delay, Stack, Subroutine, Restart, Conditional call and return instructions, Advance subroutine concepts, The 8085 Interrupts, 8085 vector interrupts.

**UNIT IV**
Program: BCD-to-Binary conversion, Binary-to-BCD conversion, BCD-to-Seven segment code converter, Binary-to-ASCII and ASCII-to-Binary code conversion, BCD Addition, BCD Subtraction, Introduction to Advance instructions and Application, Multiplication, Subtraction with carry.

**UNIT V**
8255 Programmable peripheral interface, interfacing keyboard and seven segment display, 8254 (8253) programmable interval timer, 8259A programmable interrupt controller, Direct Memory Access and 8237 DMA controller.

Introduction to 8086 microprocessor: Architecture of 8086 (Pin diagram, Functional block diagram, Register organization).

**References :**

**NCS-401: OPERATING SYSTEM**

Unit – I

Unit – II

Unit – III

Unit – IV
Memory Management: Basic bare machine, Resident monitor, Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Protection schemes, Paging, Segmentation, Paged segmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Cache memory organization, Locality of reference.

Unit – V
I/O Management and Disk Scheduling: I/O devices, and I/O subsystems, I/O buffering, Disk storage and disk scheduling, RAID. File System: File concept, File organization and access mechanism, File directories, and File sharing, Filesystem implementation issues, File system protection and security.

References:
3. Harvey M Dietel, “An Introduction to Operating System”, Pearson Education

NCS-402: THEORY OF AUTOMATA AND FORMAL LANGUAGES
Unit – I
Introduction; Alphabets, Strings and Languages; Automata and Grammars, Deterministic finite Automata (DFA)-Formal Definition, Simplified notation: State transition graph, Transition table, Language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition, Language of NFA, Equivalence of NFA and DFA, Minimization of Finite Automata, Distinguishing one string from other, Myhill-Nerode Theorem

Unit – II
Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleen’s Theorem, Regular expression to FA, DFA to Regular expression, Arden Theorem, Non Regular Languages, Pumping Lemma for regular Languages. Application of Pumping Lemma, Closure properties of Regular Languages, Decision properties of Regular Languages, FA with output: Moore and Mealy machine, Equivalence of Moore and Mealy Machine, Applications and Limitation of FA.

Unit – III
Context free grammar (CFG) and Context Free Languages (CFL): Definition, Examples, Derivation, Derivation trees, Ambiguity in Grammer, Inherent ambiguity, Ambiguous to Unambiguous CFG, Useless symbols, Simplification of CFGs, Normal forms for CFGs: CNF and GNF, Closure proper ties of CFLs, Decision Properties of CFLs: Emptiness, Finiteness and Membership, Pumping lemma for CFLs.

Unit – IV
Push Down Automata (PDA): Description and definition, Instantaneous Description, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two stack PDA

Unit – V
Turing machines (TM): Basic model, definition and representation, Instantaneous Description, Language acceptance by TM, Variants of Turing Machine, TM as Computer of Integer functions, Universal TM, Church’s Thesis, Recursive and recursively enumerable languages, Halting problem, Introduction to Undecidability, Undecidable problems about TMs. Post correspondence problem (PCP), Modified PCP, Introduction to recursive function theory

References:
6. K.Krithivasan and R.Rama; Introduction to Formal Languages, Automata Theory and Computation; Pearson Education.
NCS-403: COMPUTER GRAPHICS

Unit – I
Introduction and Line Generation: Types of computer graphics, Graphic Displays- Random scan displays, Raster scan displays, Frame buffer and video controller, Points and lines, Line drawing algorithms, Circle generating algorithms, Mid point circle generating algorithm, and parallel version of these algorithms.

Unit – II
Transformations: Basic transformation, Matrix representations and homogenous coordinates, Composite transformations, Reflection and shearing.
Windowing and Clipping: Viewing pipeline, Viewing transformations, 2-D Clipping algorithms- Line clipping algorithms such as Cohen Sutherland line clipping algorithm, Liang Barsky algorithm, Line clipping against non rectangular clip windows; Polygon clipping – Sutherland Hodgeman polygon clipping, Weller and Atherton polygon clipping, Curve clipping, Text clipping.

Unit – III
Three Dimensional: 3-D geometric primitives, 3-D Object representation, 3-D Transformation, 3-D viewing, projections, 3-D Clipping.

Unit – IV
Curves and Surfaces: Quadratic surfaces, Spheres, Ellipsoid, Blobby objects, Introductory concepts of Spline, B spline and Bezier curves and surfaces.

References:

NEC-459: MICROPROCESSOR LAB

1. To study 8085 microprocessor system
2. To study 8086 microprocessor system
3. To develop and run a programme to find out largest and smallest number
4. To develop and run a programme for converting temperature from F to C degree
5. To develop and run a programme to compute square root of a given number
6. To develop and run a programme for computing ascending/descending order of a number.
7. To perform interfacing of RAM chip to 8085/8086
8. To perform interfacing of keyboard controller
9. To perform interfacing of DMA controller
10. To perform interfacing of UART/USART

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.
NCS-451: OPERATING SYSTEM LAB

1. To implement CPU Scheduling Algorithms
   - FCFS
   - SJF
   - SRTF
   - PRIORITY
   - ROUND ROBIN
2. Simulate all Page Replacement Algorithms
   - FIFO
   - LRU
3. Simulate Paging Technique of Memory Management

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

NCS-453: COMPUTER GRAPHICS LAB

1. To implement DDA algorithms for line and circle.
2. To implement Bresenham’s algorithms for line, circle and ellipse drawing
3. To implement Mid Point Circle algorithm using C.
4. To implement Mid Point Ellipse algorithm using C.
5. To perform 2D Transformations such as translation, rotation, scaling, reflection and sharing.
6. To implement Cohen–Sutherland 2D clipping and window–viewport mapping.
7. To implement Liang Barksy Line Clipping Algorithm.
8. To perform 3D Transformations such as translation, rotation and scaling.
9. To convert between color models.
10. To perform animation using any Animation software
11. To perform basic operations on image using any image editing software
12. To draw different shapes such as hut, face, kite, fish etc.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

NCS-455: FUNCTIONAL AND LOGIC PROGRAMMING LAB

Program in SML - NJ or CAML for following:
1. To implement Linear Search.
2. To implement Binary Search.
3. To implement Bubble Sorting.
4. To implement Selection Sorting.
5. To implement Insertion Sorting.
6. Write a function that compute the factorial of a number. (factorial of 0 is 1, and factorial of n is n*(n-1)*...*1. Factorial is defined only for integers greater than or equal to 0.)
7. Write a function that evaluate a fully parenthesized infix arithmetic expression. For examples, (infix (1+(2*3))) should return 7.
8. Write a function that perform a depth first traversal of binary tree. The function should return a list containing the tree nodes in the order they were visited.
9. Write a LISP program for water jug problem.
10. Write a LISP program that determines whether an integer is prime.
11. Write a PROLOG program that answers questions about family members and relationships includes predicates and rules which define sister, brother, father, mother, grandchild, grandfather and uncle. The program should be able to answer queries such as the following:
   - father(x, Amit)
   - grandson(x, y)
   - uncle(sumit, puneet)
   - mother(anita, x)

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.