SCHEME OF TEACHING AND EXAMINATION

B.E. (ISE) III SEMESTER

Sl No.	Subject Code	Subject	Credits	Hours/Week			Examination Marks		
				Lecture	Tutorial	Practical	CIE	SEE	Total
1	UMA301C	Mathematics-III	04	4	0	0	50	50	100
2	UIS312C	Discrete Mathematical Structures	04	4	0	0	50	50	100
3	UIS303C	Logic Design	04	4	0	0	50	50	100
4	UIS314C	Computer Organization	04	4	0	0	50	50	100
5	UIS305C	Data Structures Using C	04	4	0	0	50	50	100
6	UIS307L	Logic Design Laboratory	1.5	0	0	3	50	50	100
7	UIS308L	Data Structures Laboratory	1.5	0	0	3	50	50	100
8	UIS309L	Introduction to UNIX and Shell Programming lab	2.0	0	2	2	50	50	100
9	UMA300M	Advanced Mathematics-I	--	4	--	--	50	50	100*
		Total	**25**	**20****	**2**	**8**	**400**	**400**	**800**

* Advanced Mathematics-I is a mandatory subject only for students having Diploma and admitted to 3rd Semester through lateral entry Scheme. Passing the subject is compulsory: however marks will not be considered for awarding grade/class. APP/NP grade will be awarded for passing/not passing the subject.

** The total lecture hours for students having Diploma and admitted to 3rd Semester through lateral entry scheme is 24 hours.

SCHEME OF TEACHING AND EXAMINATION

B.E. (ISE) IV SEMESTER

Sl No.	Subject Code	Subject	Credits	Hours/Week			Examination Marks		
				Lecture	Tutorial	Practical	CIE	SEE	Total
1	UMA401C	Mathematics-IV	04	4	0	0	50	50	100
2	UIS402C	Microprocessors	04	4	0	0	50	50	100
3	UIS403C	Analysis and Design of Algorithms	04	4	0	0	50	50	100
4	UIS414C	OOPs with C++	04	4	0	0	50	50	100
5	UIS415C	Operating System	03	3	0	0	50	50	100
6	UIS406C	Theoretical Foundations of Computer Science	03	3	0	0	50	50	100
7	UIS408L	Analysis of Algorithms Laboratory	1.5	0	0	3	50	50	100
8	UIS409L	OOPs Laboratory	1.5	0	0	3	50	50	100
9	UMA400M	Advanced Mathematics-II	--	4	--	--	50	50	100*
		Total	**25**	**22****	**0**	**6**	**400**	**400**	**800**

* Advanced Mathematics-II is a mandatory subject only for students having Diploma and admitted to 3rd Semester through lateral entry Scheme. Passing the subject is compulsory: however marks will not be considered for awarding grade/class. APP/NP grade will be awarded for passing/not passing the subject.

** The total lecture hours for students having Diploma and admitted to 3rd Semester through lateral entry scheme is 26 hours.

III SEMESTER

UMA301C: ENGINEERING MATHEMATICS - III
Credits (4-0-0)

Unit-I

NUMERICAL ANALYSIS:

Roots of Transcendental equations using Bisection Method, Regula-Falsi Method and Newton-Raphson. Method, Finite differences, forward, backward and central difference operators (no derivations on relations between operators), Newton-Gregory forward and backward interpolation formulae (without proof), Lagrange's and Newton's divided difference interpolation formulae (without proof). Numerical differentiation using Newton's forward & backward formulae.

Numerical Integration: Gaussian quadrature Trapezoidal rule, Simpson's one-third rule, Simpson's three eighth rule and Weddle's rule (no derivation of any formulae). Numerical solutions *of* first order ODE - Taylor's series Method, Modified Euler's method, Runge-Kutta 2^{nd} and 4^{th} order method, Milne's Predictor and *Corrector* method (problems only).

14 Hours

Unit-II

FOURIER SERIES, FOURIER TRANSFORMS, Z- TRANSFORMS:

Periodic functions, Conditions for Fourier series expansions, Fourier series expansion of continuous functions and functions having infinite number of discontinuities, even and odd functions. Half-range series, Practical Harmonic Analysis.

Infinite Fourier transforms and inverse Fourier transforms- simple properties, Complex Fourier transform, Fourier sine and Fourier cosine transforms, Inverse Fourier sine & cosine transforms, Convolution theorem & Parseval's identity (without proofs). Applications for the solutions of partial differential equations.

Z - transform - definition, standard forms, Linearity property, damping rule, shifting rule - problems.

14 Hours

Unit-III

PARTIAL DIFFERENTIAL EQUATIONS:

Formation of partial differential equations by elimination of arbitrary constants and arbitrary functions, Solution *of* equation of the type: $Pp + Qq = R$, Charpit's method, Solution of PDE by the method of separation of variables.

Derivation of one-dimensional heat and wave equations. Numerical solutions (finite difference) of One-dimensional heat and Wave equations by explicit method, Laplace equation by using standard five points formula.

12 Hours

Unit-IV

LINEAR ALGEBRA:

Rank of a matrix by elementary row transformations, Consistency of system of linear equations, Gauss elimination method, Gauss - Seidel method, characteristic values and characteristic vectors of matrices (no theorems), Largest Eigen value and the corresponding Eigen vector by Power Method.

8 Hours

CALCULUS OF VARIATIONS:

Variation *of* a function and a functional, Extremal *of* a functional, Variational Problems, Euler's equation, Standard variational problems including geodesics, minimal surface *of* revolution, hanging chain and Brachistochrone problems. **4 Hours**

TEXT BOOKS: Higher Engineering Mathematics (36th edition-2002) by Dr. B S Grewal, Khanna Publishers, New Delhi.

REFERENCE BOOKS: Advanced Engineering Mathematics by E Kreyszig (John Wiley & Sons)

UIS312C: DISCRETE MATHEMATICAL STRUCTURES
4-Credit (4-0-0)

UNIT- 1

Fundamentals
Sets and Subsets, Operations on Sets, Sequences, Division in the Integers, Matrices, Mathematical Structures

Logic: Propositions and Logical Operations, Conditional Statements, Methods of Proof, Mathematical Induction

UNIT- 2

Counting: Permutations, Combinations, The Pigeonhole Principle, Elements of Probability, Recurrence Relations

Relations and Digraphs
Product Sets and Partitions, Relations and Digraphs , Paths in Relations in Digraphs , Properties of Relations , Equivalence Relations, Computer Representation of Relations and Digraphs, Operations on Relations, Transitive Closure and Warshall's Algorithm

UNIT- 3

Functions: Functions, Functions for Computer Science, Growth of Functions, Permutation Functions

Order Relations Partially Ordered Sets, Extremal Elements of Partially Ordered Sets, Lattice

Semi Groups and Groups: Binary Operations Revisited, Semi groups, Products and Quotients of Semi groups and Groups, Products and Quotients of Groups

UNIT- 4

Trees: Trees, Labeled Trees, Tree Searching, Undirected Trees, Minimal Spanning Trees

Topics in Graph Theory: Graphs, Euler Paths and Circuits, Hamiltonian Paths and Circuits, Transport Networks, Matching Problems, Coloring Graphs

Text Book:

1.Kolman, Busby, Ross, "Discrete Mathematical Structures", 5th edition, PHI.

Reference Books:

1. Gary Haggard, John Schlipf, Sue White sides, "Discrete Mathematics for Computer Science", Thomson Publications.

2. Ralph P Grimaldi, "Discrete and Combinatorial Mathematics-An applied Introduction", Pearson Education, 4th Edition.

UIS303C: LOGIC DESIGN
4-Credit (4-0-0)

UNIT I

Boolean algebra and Combinational networks
Definition of Boolean algebra, Principle of duality, Boolean formulas and functions, Canonical formulas, Manipulation of Boolean formulas and combinational circuits, Incomplete Boolean functions and Don't care conditions, Additional Boolean operations and gates, Introduction to HDL

Simplification of Boolean expressions
Formulation of the Simplification problem, Prime implicants and Irredundant Disjunctive expressions, Prime implicates and irredundant conjunctive expressions, HDL implementation for these Irredundant functions.

12 Hrs

UNIT II

Simplification of Boolean expressions
Using K-map to obtain minimal expression, Minimal expressions for incomplete functions, The Quine-McClauskey method of generating prime implicants and prime implicates, Prime implicants / prime implicates tables and Irredundant expressions, Decimal method for obtaining prime implicants, Variable Entered K-map, HDL implementation of logic circuits

Logic Design with MSI components
Binary adders and subtracters, Decimal Adders

12 Hrs

UNIT III

Logic Design with MSI components
Comparators, Decoders, Encoders, Multiplexers

Programmable logic Devices
Programmable Logic Devices, PROM, Programmable Logic Array, Programmable Array Logic, HDL implementation.

Flip-Flops and applications
Basic bistable element, Latches, Master Slave Flip-Flops, Edge Triggered Flip-Flops, Characteristic equations, registers

14 Hrs

UNIT IV

Flip-Flops and applications
Counters, Design of Synchronous Counters, HDL implementation,

Synchronous Sequential networks
Structure and operation of clocked Synchronous sequential networks, Analysis of clocked Synchronous sequential networks

D/A Conversion and A/D conversion
Variable, resistor networks, Binary Ladders, D/A converters, D/A accuracy and resolution, A/D converter- Simultaneous conversion, A/D converter – Counter Method

14 Hrs

Text Books.

1. Digital Principles and design- Donald D. Givone (TMH)
2. Digital Principles and applications – Albert Paul Malvino, Donald P Leach and Goutam Saha (TMH), 6th Edition (chapters 2.5, 3.11, 4.14, 6.128.12, 9.7, 10.9, 12)

Reference Books

1. Fundamentals of Digital Logic with Verilog Design, Stephen Brown, Zvonko Vranesic, TMH, 2006
2. Fundamentals of Logic Design, Charles H. Roth, Jr., 5th Edition,Thomson, 2004
3. Digital Systems Principles and Applications, Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, 10th Edition, PHI/Pearson Education, 2007

UIS305C: DATA STRUCTURES USING C
4 CREDITS (4-0-0)

UNIT I

Pointers & its applications: Concepts, pointer variables, accessing variables through pointers, pointer declaration & definition, initialization of pointer variables, pointers & functions, pointers to pointers, lvalue & rvalue , pointer applications, arrays & pointers, pointer arithmetic and arrays, memory allocation functions, array of pointers.
4 hrs

Strings: String concepts, C strings, string input/output functions, arrays of strings, string manipulation functions. **2 hrs**

Derived types – Enumerated, Structure and Union: The type definition (typedef), Enumerated type definition, Structure, accessing structures, complex structures, array of structures, structures and functions, unions. **4 hrs**

Text files: Concept of file, files & streams, standard library input/output functions, character input/output functions, character input/output examples. **3 hrs**

UNIT II

The stack: ***Definition and Examples:*** Primitive operations, The stack as an Abstract data type, ***Representing Stacks in C***: Implementing pop operation, Testing for Exceptional Conditions, Implementing the push operations. ***An Example- Infix, Postfix and Prefix*** : Evaluating a postfix Expression, Program to evaluate a postfix Expression, Limitations of the program, converting an Expression from Infix to Postfix, program to convert an Expression from Infix to Postfix.
6 hrs

Recursion: ***Recursive definition and processes***: The factorial function, Multiplication of Natural numbers, The Fibonacci sequence, The Binary search, properties of recursive definitions or Algorithms. ***Recursion in c:*** Factorial in C, The binary search in C, Recursive chains, Recursive definitions of algorithmic Expressions, ***Rewriting recursive programs***: The Towers of Hanoi Problem. **4 hrs**

Queues: ***The queue and its sequential representation***: The queue as an abstract data type, C implementation of queue, the insert operation, the priority queue, Array implementation of a priority queue**.** **3 hrs**

UNIT III

Lists: ***Linked lists***: Inserting and removing nodes from a list, linked implementation of stacks, the getnode and freenode operations, linked implementation of queues, the linked list as a data structure, examples of list operations, list implementation of priority queues, ***lists in c***: array implementation of lists, limitations of the array implementation, allocating and

freeing dynamic variables, linked lists using dynamic variables, queues as lists in c, non integer and nonhomogeneous lists, comparing the dynamic and array implementation of lists. **8 hrs**

Other list structures: Circular lists, the stack as a circular lists, the queue as a circular list, primitive operations on circular lists. **5 hrs**

UNIT IV

Trees: ***Binary trees:*** operation on Binary trees, Applications of Binary Tree, ***Binary tree representation****:* Node representations of Binary Trees, Internal & External Nodes, Implicit array representation of binary trees. Choosing a binary tree representation, binary traversal in c. Threaded Binary trees. Traversal using father field, Heterogeneous Binary Trees. ***Trees and their applications*** : C representation of trees, tree traversals, general expressions as trees, evaluating an expression tree, constructing tree**.** **8 hrs**

Sorting: ***General background***: Efficiency considerations, O notation, efficiency of sorting, ***Selection and tree sorting***: straight selection sort.
Searching: ***Basic search techniques:*** the dictionary as an Abstract data type, algorithm notation, Sequential searching, Efficiency of sequential searching, Recording a list for maximum search efficiency, searching an ordered table, the indexed sequential search, the binary search, interpolation search. **5 hrs**

Text books:

1) Behrouz A. Forouzan, Richard F. Gilberg , "A Structured Programming Approach Using C", Second Edition, Thomson Brooks/Cole .
2) Aaron M. Tennenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data structure using C ", Pearson Education/PHI 2006.

Reference Books:

1) Behrouz A. Forouzan and Richard F. Gilberg, Thomson, "Computer Science A structured Programming Approach using C " II edition, 2003.
2) Richard F. Gilberg and Behrouz, Data structures "A pseudo code approach with c " , Thomson, 2005.
3) Robert Kruse and Breuse Leung," Data structures and program Design in C", PEARSON Education, 2007

UIS314C: COMPUTER ORGANIZATION
4-CREDIT (4-0-0)

Unit I

Basic Structure of Computer**:** Computer Types. Functional Units, Basic Operational Concepts, Bus Structures, Performance – processor clock, Basic Performance Equation, Clock rate, Performance Measurement, Machine Instructions and Programs**:** Numbers, Arithmetic Operations and Characteristics, Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing. Addressing Modes, Assembly language, Basic Input and Output operations, Stacks and Queues, Subroutines.

13 HRS

Unit II

Additional Instructions, Encoding of machine instructions. Input/Output organization**:** Accessing I/O Devices, Interrupts-interrupt hardware, Enabling and disabling interrupts, Handling multiple devices, Controlling device requests, Exceptions, Direct memory access, Buses, Interface circuits, Standard I/O interfaces-USB; Device characteristics, Architecture, Addressing.

13 HRS

Unit III

Memory system**:** Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size and cost, Cache Memories, Mapping Functions, Virtual memories, Basic Arithmetic concepts for ALU**:** Addition and subtraction of signed numbers, Design of fast adders; Carry-lookahead addition only, Multiplication of positive numbers,.

13 HRS

Unit IV

Signed operand multiplication, Fast multiplication, Integer division,Floating point numbers and operations, Basic processing unit: Fundamental concepts, Execution of a complete instruction, Multiple bus organization, Hard-wired control, Micro programmed control.

13 HRS

Text Book:

1. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, 5th edition, TMH, 2002.

Reference Books:

1. Computer Organization and Architecture, William Stallings,7th edition,PHI,2006
Computer Systems Design and Architecture, Vincent P.Heuring & Harry F.Jordan, 2nd edition, Pearson Education, 2004.

UIS307L: LOGIC DESIGN AND COMPUTER HARDWARE LABORATORY
1.5 CREDITS (0-0-3)

Logic Design Lab

1. Realization of a given Boolean Expression using MEV method.
2. Design and implementation of BCD to Excess-3 using 4-Bit Adder Chip and Logic Gates.
3. Design and implement Full adder using 3:8 Decoder (74138) and Simulate full adder.
4. Design ad implement Full subtracter using 8:1 multiplexer (74154) and simulate the 8:1 Multiplexer.
5. Design JK master/slave FF using NAND gates and simulate.
6. Design and implementation of 3 bit Mod-n synchronous counter using JK flip-Flops(7476) (where n<=8) and simulate
7. Design and implementation of Ring counter and Johnson counter using 4-bit shift register and simulate
8. Design and implementation of an Asynchronous Counter using a Decade Counter IC to count up from 0 to n (n<9). Display the count value on 7-segment LED display using BCD to 7-segment code converter IC.
9. Design of 4-bit R-2R ladder DAC using Op-Amp.

Hardware lab

1. PC Assembly & Dis-assembly
2. Demonstration of Bus operation
3. Storage Device swapping.
4. Parallel port, serial port and USB port testing
5. Understand BIOS and CMOS
6. Understand Chipset
7. Backup and Restore techniques
8. Data Recovery techniques

Reference Books

1. Trouble shooting your PC --- M. David stone and alfred poor PHI edition
2. Enhanced guide to managing and maintaining your PC Jean Andrews, Thomson course technology, III edition
3. IBM PC & CLONES Fifth Edition EBPB - Rajesh Hongal

UIS308L: DATA STRUCTURE LAB
1.5 CREDITS (0-1-2)

Programs on pointers and their usage:

1) Write a C program to compute roots of a quadratic equation. Use pointers to pass data from a read function, pass both values and pointers to a compute function, and finally pass the values to a print function.

Programs on strings:

2) Write a C program that converts a string representing a number in Roman numeral form to

   decimal form.

Programs on structures and unions:

3) Write a C program that uses an array of student structures to answer inquiries. Using menu
   driven user interface, provide inquiries as below,
   The first menu option accepts student information such as name, USN, marks for 5 subjects,
   second option report a student USN, marks, average & grade based on absolute scale of 90%
   for A, 80% for B, 70% for C, 60% for D and score below 60% is an F & third option provides all data for requested student.

Programs on File handling in C:

4) Create the employee file shown below

Employee No.	Department	Pay Rate	Hours Worked
Positive Integer	10 Characters	Floating point number	Positive Integer

   Write a C program to read the employee file and create a payroll register. The register will contain the following data: Employee number, department, pay rate, Hours worked, Basic pay (pay rate * hours worked).

Programs on stack:

5) write a C program to simulate the working of Coin Tube (which is top quality and protect coins from moisture) and to perform the following operations on it:
   a. Insert coin
   b. Remove coin
   c. Display all coins of coin tube.

   The program should print appropriate messages for *coin tube overflow*, *coin tube underflow*,

and *coin tube empty*.

6) Write a C Program to convert and print a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).

Program on recursion:

7) Write recursive C Program for solving the Towers of Hanoi problem.

Programs on queues:

8) Write a C program to simulate the working of Messaging System in which a message is
placed in a *Queue* by a *Message Sender,* a messag*e* is removed from the queue by a *Message*
Receiver, which can also display the contents of the *Queue*

9) Write a C Program to simulate the working of a Circular queue of integers using an array. Provide the following operations:
   a. Insert
   b. Delete
   c. Display

Programs on linked lists:

10) Write a C Program using dynamic variables and pointers, to construct a singly linked list consisting of the following information in each node: student id (integer), student name (character string) and semester (integer). The operations to be supported are:

    a. The insertion operation
    i. At the front of a list
    ii. At the back of the list
    iii. At any position in the list

    b. Deleting a node based on student id. If the specified node is not present in the list an error message should be displayed. Both the options should be demonstrated.
    c. Searching a node based on student id and update the information content. If the specified node is not present in the list an error message should be displayed. Both situations should be displayed.
    d. Displaying all the nodes in the list.

11) Write a C Program to support the following operations on a doubly linked list where each node consists of integers:
    a. Create a doubly linked list by adding each node at the front.
    b. Insert a new node to the left of the node whose key value is read as an input
    c. Delete the node of a given data, if it is found, otherwise display appropriate message.

d. Display the contents of the list.

Program on trees:

12) Write a C Program to construct a Binary tree & implement tree traversal methods.

Program on sorting:

13) Write a C program to sort a given integer elements by using Selection sort.

Group Tasks:

1. Mini search Engine
2. Word Transformation
3. Expression Evaluation
4. Car garage system simulation
5. Dictionary
6. Simulating calculator

General remarks:

1) Lab schedule: 3hrs/week for each student (1 hr tutorial, 2 hrs- program execution).
2) Student should complete all the lab assignments.
3) One mini project adequate per student batch (students/batch).
4) Each student should demonstrate his/her group work individually.
5) Marks Evaluation:

   Lab assignments : 40%

   Group work : 10%

Note: Develop above mentioned group work using any of the data structures.

UIS309L: INTRODUCTION TO UNIX AND SHELL PROGRAMMING LAB
2 Credits (0-2-2)

INTRODUCTION TO UNIX

UNIT I

1. **BACKGROUND AND SOME BASIC COMMANDS** **4 hrs**

Brief history and Salient features of a Unix System, POSIX and the Single UNIX specification, The UNIX Architecture, Locating Commands. Internal and External Commands, Flexibility of Commands Usage, **man**: Browsing and MANUAL pages On-line, Understanding the man Documentation.

cal: The calendar, **date**: Displaying and System Date, **echo**: Displaying a message**, printf:** An Alternative to echo, **bc**: The Calculator, **script**: Recording Your Session, **passwd**: Changing Your Password, **who**: Who Are Users?, **uname**: Knowing Your Machine's Characteristics, **tty:** Knowing Your Terminal, **stty**: Displaying and Setting Terminal Characteristics.

2. THE FILE SYSTEM AND SOME FILE HANDLING COMMANDS **9 hrs**

The File, content of the file, The Parent and Child Relationship, The HOME Varible: The Home Directory, pwd: Checking Your Current Directory, cd: Changing the Current Directory, mkdir: Making Directories, rmdir: Removing Directories, Absolute Pathnames, Relative Pathnames, ls: Lising Directory Contents, The UNIX File System

cat: Displaying and Creating Files, cp: Copying a File, rm: Deleting Files, mv: Renaming Files, more: Paging Output, The lp Subsytem: Printing a Characters, od: Displaying Data in octal, FILE ATTRIBUTES: ls –l: Listing File attributes, The –d option: Listing Directory Attributes, listing inode number, listing hidden files, time associated with a file, listing timestamps, File Ownership, File Permissions, chmod: Changing File Permissions, Directory Permissions, Changing File Ownership, File System and Inodes, Hard Links, Symbolic Links and ln, The directory, umask: Default File and Directory Permissions, Modification and Access Times, find: Locating Files.

UNIT II

4 THE SHELL AND THE PROCESS **3 hrs**

The shell's Interpretive Cycle, Pattern Matching- The Wild-cards, Escaping and Quoting, Redirection: The Three standard Files, /dev/null and /dev/tty: Two Special Files, Pipes, tee: Creating a Tee, Command Substitution, Shell Variables

Process Basics, ps: Process Status, System Processes (-e or –a), Mechanism of Process creation, Internal and External Commands, Running Jobs in Background, nice: Job execution with low priority, Killing Processes with Signals, Job Control, at: and batch: Execute Later, cron: Running Jobs Periodically, time: Timing Processes

5. SIMPLE FILTERS AND grep FAMILY OF COMMANDS **3 hrs**

The Sample Data base, pr: Paging Files, head: Displaying the Beginning of a File, tail: Displaying the End of a File, cut: Slitting a File Vertically, paste: Pasting Files, sort: Ordering a File, Uniq; Locate Repeated and Non repeated Lines, tr: Translating Characters, An Example : Displaying a word count List, grep: Searching for a Pattern, Basic Regular Expressions- An Introduction, Extended Regular Expressions and egrep

6. SHELL PROGRAMMING **7 hrs**

Environment Variables, Aliases (bash and ksh), Command History (bash and ksh), Shell Scripts, read and read only commands, Using Command line Arguments, exit and exit status of the command, The Logical Operators && and ||- Conditional Execution. The if Conditional, Using test and [ ] to Evaluate Expressions, The case Conditional, expr: Computation and String Handling, $0: Calling a Script by Different names, while: Looping, for: Looping with a List, set and shift: Manipulating the Positional Parameters, The here Document (<<), trap: Interrupting a Program, Debugging Shell Scripts with set –x, export: Exporting Shell Variables, eval: Evaluating Twice.

Text Books:

1. Sumitaba Das, Unix Concepts and Applications, Third edition Tata McGraw Hill 2007

Reference Books:

1. .Behrouz A. Forouzan and Richard F. Gilberg, Unix and Shell Programming, A Text Book, Thomson, Edition-2003.
2. Ellie Quigley, Unix Shells By Examples, Pearson Edition, Fourth Edition
3. Brain W. Kernighan and Rob Pike, The Unix Programming Environment Pearson Education, Edition -2006

UNIX SHELL PROGRAMMING LAB

1. Write a shell script called greetme that will do the following: Greet the user, print the date and the time, print a calendar for this month, print the name of your machine, print the name and release of this operating system, print a list of all files in your parent directory, print all the processes root is running, print the value of the TERM, PATH, and HOME variables, print your disk usage, print your group id.

2. Write a shell script called info that will do the following: ask the user's full name-first name, last name and middle name, greet the user by his or her first name, ask the user's year of birth and calculate his or her age, ask the user's login name and print his or her user id, tell the user his or her home directory, show the user processes he or she is running, tell the user the day of the week, and the current time.

3. Create a text file called datafile. Each entry consists of fields separated by colons. The fields are as follows: first name and last name, phone number, address, birth date, salary. Create a script called lookup that will do the following: sort datafile by last names, show the user the contents of datafile, tell the user the number of entries in the file

4. Write a shell script to find any files in the root partition that have not been modified within the past n days and are larger than 20 blocks(512 byte blocks)

5. Write a shell script that accepts valid file names as command line arguments and for each of the arguments, prints the type of the file (regular file, directory file character device file, block device file, symbolic link file etc and also print permission on file.

6. Write a shell script to find and display all the links of a file specified as the first argument to the script. The second argument, which is optional, can be used to specify the directory in which the search is to begin. If this second argument is not present, the search is to begin in current working directory. In either case the starting directory as well as its subdirectories at all levels must be searched. The script need not include any error checking.

7. Write a shell script to provide nicer interface to cal command without changing cal itself. It should recognize month by name. With two arguments, it should behave just as the old cal does, except for converting month names into numbers. Given one argument, it should print the month or year's calendar as appropriate, and given zero arguments, it should print the current month's calendar.

8. Write a shell script to report which file correspond to a command is executed.

9. Write a shell script to implement getopts statement, your script should understand following command line arguments and called this script Q14 –c –d –e where options work as
   -c clear the screen
   -d show list of files in current working directory

10. Write a shell script called sayHello, put this script into your startup file called .bash_profile, the script should run as soon as you logon to system, and it print any one of the following message in infobox using dialog utility, if installed in your system, if dialog utility is not installed then use echo statement to print message: Good morning, Good afternoon, Good evening, according to system time

UMA300M: ADVANCED MATHEMATICS-I
(MANDATORY SUBJECT)
(COMMON TO ALL BRANCHES)

1) Trigonometry:
Complex Numbers: Definitions, complex numbers as an ordered pair, real and imaginary parts, modulus and amplitude of a complex number, equality of a complex number, addition, subtraction, multiplication & division of complex numbers, polar form, Argand Diagram, exponential form, expressing in the form a + ib problems.

06 Hours

2) Differential Calculus:
Differentiation of nth order of standard functions, Leninist's theorem (Statement only) with examples, polar curves, Taylor's series, Maclaurian's series of simple functions for single variable
Partial Differentiation: Definition, Euler's theorem, total differentiation, Differentiation of composite and implicit functions, jacobians illustrative examples and problems

14 Hours

3) Integral Calculus:
Reduction formula for functions $Sin^n x$, $Cos^n x$, $Sin^m x Cos^n x$. Double integral, simple problems & Triple integral simple problem(with standard limits), β and γ functions, properties, relation between β and γ functions simple problems

08 Hours

4) Differential Equations:
Solution of first order, first-degree differential equations – variable separable methods, homogeneous equation, Bernoulli's and exact differential equations (without I.F.) Differential equations of second and higher orders with constant coefficients

12 Hours

Text Books:

1) Higher Engineering Mathematics – B S Grewal 36^{th} edition, Khanna Publisher
2) Higher Engineering Mathematics – H K Das 25^{th} Edition, S.Chand and sons publications

Question Paper Pattern:

1. Total of Eight Questions to be set uniformly covering the entire syllabus.
2. Each question should not have more than 4 sub divisions.
3. Any five full questions are to be answered.

IV SEMESTER

UMA401C: ENGINEERING MATHMATICS
CREDITS (4-0-0)

Unit-I

COMPLEX VARIABLES:

Analytic functions, Cauchy-Reimann equations in Cartesian and Polar forms - consequences, construction of analytic function (Cartesian and polar forms), Definition of Conformal transformations: Z^2, e^z and $(z + a^2/z)$ $(z \neq 0)$ Bilinear transformations.

Complex Integrations: Line integral, Cauchy's theorem - corollaries, Cauchy's integral formula. Taylor and Laurent's series (statements only), Singularities, Poles, Calculation of Residues, Residue theorem (Without proof) - problems. Contour Integration.

14 Hours

Unit-II

SPECIAL FUNCTIONS:

Series solution of ordinary differential equation about ordinary point and regular singular point, Hypergeometric, Hermite, Legendre, Bessels and Chebeschev equations. Recurrence formulae, Generating function, orthogonal property, Rodrigue's formula.

14 Hours

Unit-III

STATISTICS AND PROBABILITY:

Curve fitting by the method of least squares: $y = a+bx$, $y = ab^x$, $y = a+bx+cx^2$. Correlation and Regression. Probability - addition rule, conditional probability, multiplication rule, Baye's' rule. Discrete and continuous random variables-PDF-CDF, Binomial, Poisson and Normal distributions **12 Hours**

Unit-IV

SAMPLING DISTRIBUTION:

Sampling, Sampling distribution, Standard error, Null and alternate hypotheses, Type I and Type II errors, Testing of hypothesis for Means, Level of Significance for Means, Confidence limits for Means, large and small samples, Student's *t*-distribution.
Central limit theorem (without proof)

JOINT ROBABILITY DISTRIBUTION AND MARKOV CHAINS:

Concept of joint probability, Joint distributions - discrete random variables, Independent random variables, Markov chains, higher transition probabilities, stationary distribution of regular Markov chains and absorbing states.

12 Hours

TEXT BOOKS: 1) Higher Engineering Mathematics (36th edition-2002) by Dr. B S Grewal, Khanna Publishers, New Delhi.

2) Theory and Problems of Probability by Seymour Lipschutz (Schaum' Series) - Relevant articles of Chapter 5 and Chapter 7.

REFERENCE BOOKS: Advanced Engineering Mathematics by E Kreyszig (John Wiley & Sons)

UIS402C: MICROPROCESSORS
4 CREDITS (4-0-0)

UNIT I

An Introduction to Computers, Microcomputers, and Microprocessor
Types of computers; Overview of microcomputer structure and operation; Microprocessor evolution and types; The 8086 family overview-8086 Internal architecture; Introduction to programming 8086

Introduction to 8086 Family Assembly Language Programming
Program development steps; Writing programs for use with an assembler; Assembly language program development tools; Constructing machine codes for 8086 instruction; MOV instruction coding examples

Implementing Standard Program Structure In 8086 Assembly Language
Simple sequence programs; Jumps, Flags, and Conditional Jumps; If-Then, If-Then-Else, and Multiple If-Then-Else programs; While-Do programs; Repeat-Until Programs; Instruction timing and delay loops

13 Hours

UNIT II

8086 Instruction Description and Assembler Directives
Instruction descriptions; Assembler directives

Strings, Procedures, and Macro
The 8086 string instructions; Writing and using procedures; Writing and using assembler macros.

13 Hours

UNIT III

Using Assembly Language with C/C++
Using assembly language with C/C++ for 16-bit applications.

8086 Interrupts and Interrupt Applications
8086 Interrupts and interrupt responses; Hardware Interrupt applications; 8254 software-programmable timer/counter; 8259A priority interrupt controller; software interrupt applications.

8086 System Connections and Timing
A basic 8086 microcomputer system; 8086 minimum and maximum modes; Addressing memory and ports in microcomputer systems; 8086 addressing and address decoding; How the 8086 microprocessor accesses memory and ports; 8086 timings.

13 Hours

UNIT IV

Digital Interfacing and Analog Interfacing
Programmable parallel ports and handshake input/output; Interfacing microprocessor to keyboards; Interfacing to alphanumeric displays; Review of operational amplifier

characteristics and circuits; Sensors and Transducers; D/A operation, interfacing; A/D specifications, interfacing

Direct Memory Access, Coprocessors
Basic DMA operation; The 8237 DMA controller; A Coprocessor- The 8087 math coprocessor.

The Pentium II, Pentium III, and Pentium IV Microprocessors
Introduction to Pentium II microprocessor; Pentium II software changes; The Pentium III; The Pentium IV.

13 Hours

Note: Every student has to do one assignment comparing atleast 4 processors belonging to different families.

Text Books:

1. **Microprocessor and Interfacing,** Douglas V. Hall, Revised 2nd Edition, TMH, 2006.(Chapters 2, 3,4,5,6,7,8,9,10,11)

2. **The Intel Microprocessors,** 6th Edition, Barry B. Brey, Pearson/PHI 2006.(Chapters 8,13, 19)

Reference Books:

1. **Advanced Microprocessors and IBM-PC assembly Language programming,** K. Udaya Kumar and B.S. Umashankar, TMH 2003.
2. **The Intel Microprocessor Family: Hardware and Software Principles and Applications,** James L. Antonakos, Thomson, 2007.

UIS403C: ANALYSIS AND DESIGN OF ALGORITHMS
4 CREDITS (4-0-0)

UNIT I

Introduction: What is an Algorithm?, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental Data Structures **6 Hours**

Fundamentals of the Analysis of Algorithm Efficiency: Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Nonrecursive and Recursive Algorithms, Example –Fibonacci Numbers
7 Hours

UNIT II

Brute Force: Selection Sort and Bubble Sort, Sequential Search and Brute-Force String Matching, Exhaustive Search **3 Hours**

Divide and Conquer: Mergesort, Quicksort, Binary Search , Binary tree traversals and related properties, Multiplication of large integers and Stressen's Matrix Multiplication. **5 Hours**

Decrease and Conquer: Insertion Sort, Depth First Search, Breadth First Search, Topological Sorting, Decrease by constant factor Algorithms,Variable –size-decrease algorithms. **5 Hours**

UNIT III

Transform and Conquer: Presorting, Balanced Search Trees, Heaps and Heapsort, Problem Reduction **5 Hours**

Space and Time Tradeoffs: Sorting by Counting, Input Enhancement in String Matching, Hashing, B-trees.
4 Hours

Dynamic Programming: Computing a Binomial Coefficient, Warshall's Algorithm, : Floyd's Algorithms, The Knapsack Problem and Memory Functions **4 Hours**

UNIT IV

Greedy Technique: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees, Knapsack Problem.
4 Hours

Limitation of Algorithm Power: Lower-Bound arguments, Decision Tree,P,NP,NP-Complete Problem. **4 Hours**
Coping with the Limitations of algorithm power : Backtracking And Branch & Bound techniques Approximation Algorithm for NP-Hard problems. **5 Hours**

Text Book

1. **Introduction to The Design & Analysis of Algorithms** , Anany Levitin, 2nd Edition, Pearson ducation, 2007. (Chapter 1, 2.1 to 2.5, 3.1, 3.2, 3.4, 4.1 to 4.5, 5.1 to 5.4, 6.1, 6.3, 6.4, 6.6, 7.1 to 7.3, 8.1, 8.2, 8.4, 9, 11.1, 11.2, 11.3, 12.1, 12.2, 12.3).

Reference Books

1. **Introduction to Algorithms** , Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 2nd Edition, PHI, 2006.
2. **Computer Algorithms** by Horowitz E., Sahni S., Rajasekaran S., Galgotia Publications, 2001.

UIS414C: OBJECT ORIENTED PROGRAMMING WITH C++
4 CREDITS (4-0-0)

UNIT I

Principles of Object Oriented Programming: Software Evolution, A look at procedure-oriented Programming, Object-oriented programming paradigm, Basic concepts of OOP, Object Oriented languages, Applications of OOP.

Introduction to C++: Console Input/Output in C++, Variables in C++, Reference Variables in C++, Function Prototyping, Function Overloading, Default Values for Formal Arguments of Functions, Inline Functions

Classes and Objects: Introduction to Classes and Objects, Member Functions and Member Data, Objects and Functions, Objects and Arrays, Namespaces, Nested Classes

12 Hours

UNIT II

Constructors and Destructors: Constructors, Destructors, the Philosophy of OOPS

Inheritance: Introduction to Inheritance, Base Class and Derived Class Pointers, Function Overriding, Base Class Initialization, The Protected Access Specifier, Deriving by Different Access specifiers, Different kinds of Inheritance, Order of invocation of constructors and destructors.

Dynamic Memory Management: Introduction, Dynamic Memory Allocation, Dynamic Memory Deallocation

13 Hours

UNIT III

Operator Overloading: Operator Overloading, Overloading the Various Operators - Overloading the Increment and the Decrement Operators (Prefix and Postfix), Overloading the Unary Minus and the Unary Plus Operator, Overloading the Arithmetic Operators, Overloading the Relational Operators , Overloading the Assignment Operator, Overloading the Insertion and Extraction Operators, Overloading the new and the delete Operators, Overloading the Subscript Operator, Overloading the Pointer-to-member Operator (->)

13 Hours

UNIT IV

Virtual Functions and Dynamic Polymorphism: The Need for Virtual Functions, Virtual Functions, The Mechanism of Virtual Functions, Pure Virtual Functions, Virtual Destructors and Virtual Constructors

Templates: Introduction, Function Templates, Class Templates, The Standard Template Library (STL)

Stream Handling: Streams, The Class Hierarchy of Handling Streams, Text and Binary Input/Output, Text Versus Binary Files, Text Input/Output, Binary Input/Output, Stream Handling contd.: Opening and Closing Files, Files as Objects of the fstream Class

Type Conversion: Type Conversion, New Style Casts, and RTTI

Exception Handling: Introduction, C-Style Handling of Error-generating Codes, C++ Style Solution - the try/throw/catch Construct, Limitation of Exception Handling

14 hours

TEXT BOOKS

1. Object-Oriented Programming with C++, Sourav Sahay, Oxford University Press, 2006 Chapter 1 (1.5 - 1.11), Chapter 2, 3(3.1 – 3.3), 4, 5, 6, 7 (7.1 – 7.8), 8, 9, 10

2. Object-Oriented Programming with C++ by E. Balaguruswamy. Tata McGraw Hill - II Edition Chapter 1(1.2 - -1.8)

REFERENCE BOOKS

1. The Complete Reference C++ by Herbert Schildt, Tata McGraw Hill - IV Edition

2. Object-Oriented Programming with C++ by P.B.Kotur

UIS415C: OPERATING SYSTEM
3 Credits (3-0-0)

UNIT 1

INTRODUCTION TO OPERATING SYSTEMS, PROCESS MGMT

Role of Operating systems: user view, system view; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and security; Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating System design and implementation; Operating System structure; Virtual machines.

Process management: Process concept; Operations on processes; Process Scheduling: Basic concepts; scheduling criteria; Scheduling algorithms; Multiple-Processor scheduling.

10 Hrs

UNIT 2

THREADS AND PROCESS SYNCHRONIZATION

Interprocess communication, Threads: concepts, Multi-Threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Thread scheduling.

Synchronization: The Critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.

10 Hrs

UNIT 3

DEADLOCKS AND MEMORY MANAGEMENT

Deadlocks: Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

Memory Management Strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. Virtual Memory Management: Background; Demand paging; Page replacement; Allocation of frames; Basics concepts of Thrashing.

10 Hrs

UNIT 4

FILE SYSTEM: CONCEPTS AND IMPLEMENTATION, SECONDARY STORAGE STRUCTURES

File system: File System: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection. Implementing File System: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.

Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management, Protection: Goals,principles and domain of protection. 10 Hrs

Text Books:

Abraham Silberschatz, Peter Baer Galvin , Greg Gagne: Operating System Principles, 7th edition, Wiley-India, 2006.

Reference Books:

1. D.M Dhamdhere: Operating systems - A concept based Approach, 2nd Edition, Tata McGraw- Hill, 2002.
2. P.C.P. Bhatt: Operating Systems, 2nd Edition, PHI, 2006.
3. Harvey M Deital: Operating systems, 3rd Edition, Addison Wesley, 1990.

UIS406C: THEORETICAL FOUNDATIONS OF COMPUTER SCIENCE
3 CREDITS (3-0-0)

Unit-I

Introduction to Finite Automata: Introduction to Finite Automata; The central concepts of Automata theory; Deterministic finite automata; Nondeterministic finite automata.

Finite Automata, Regular Expressions: An application of finite automata; Finite automata with Epsilon-transitions; Regular expressions; Finite Automata and Regular Expressions; Applications of Regular Expressions. **11Hours**

Unit-II

Regular Languages, Properties of Regular Languages: Regular languages; Proving languages not to be regular languages; Closure properties of regular languages; Decision properties of regular languages; Equivalence and minimization of automata.

Context-Free Grammars And Languages: Context–free grammars; Parse trees; Applications; Ambiguity in grammars and Languages.
11Hours

Unit-III

Pushdown Automata: Definition of the Pushdown automata; The languages of a PDA; Deterministic Pushdown Automata.

Properties of Context-Free Languages: Normal forms for CFGs; The pumping lemma for CFGs; **10Hours**

Unit-IV

Introduction To Turing Machine: Problems that Computers cannot solve; The turning machine; Programming techniques for Turning Machines; Extensions to the basic Turning Machines; Turing Machine and Computers. **10Hours**

Text Books:

1. John E.. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman: Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson education, 2007. (Chapters: 1.1, 1.5, 2.2 to 2.5, 3.1 to 3.3, 4, 5, 6.1, 6.2, 6.4, 7, 8.1 to 8.4, 8.6)

Reference Books:

1. Peter. Linz: An Introduction to Formal Languages and Automata", Third Edition, Fifth printing
2. Raymond Greenlaw, H.James Hoover: Fundamentals of the Theory of Computation,
1. Principles and Practice, Morgan Kaufmann, 1998.
2. John C Martin: Introduction to Languages and Automata Theory, 3rd Edition, Tata McGraw-Hill, 2007.
3. Daniel I.A. Cohen: Introduction to Computer Theory, 2nd Edition, John Wiley & Sons, 2004.
4. Thomas, A., Sudkamp: An Introduction to the Theory of Computer Science, Languages and Machines, 3rd Edition, Pearson Education, 2006
5. John, E., Hopcroft, Jeffrey D.Ullman: “Introduction to Automata Theory, Languages and Computation”, Narosa.

UIS408L: ANALYSIS OF ALGORITHMS LABORATORY
1.5 CREDITS (0-0-3)

1. Consider the following standard time complexity functions:
   F(n)=n, log n , n2, n3, 2n , n!
   Write a program to that takes input value 'n' and find the value of F(n) by varying I from 1 to n Note : Use switch statement to consider different functions.
   Draw the graph of **Time versus 'Input n'** to display these results, and observe the effect of increasing the n value for each function.

2. Write a program to generate 'n' random numbers and sort them using bubble sort. Perform apriority and posteriori analysis (time complexity) in following cases.
   a. n= 5, and the value of each number is lesser than 100
   b. n=10000, and the value of each number is lesser than 10000
   Analyze the memory requirements in each case.

3. Write a program for recursive binary search and Iterative Binary search in an a array of N elements. Perform apriority and a posteriori analysis of the time required to search an element in following cases.
   a. N=10 (Manually Entering inputs).
   b. N=5000.
   c. N= 50000.

4. How many comparisons (both successful and unsuccessful) will be made by the brute-force string-matching algorithm in searching for each of the following pattern in the binary text of 1000 zeros and/or ones ?
   a. 00001 b. 10000 c. 01010

5. Apply Quick sort method to sort elements of array in following cases. Observe the best case and worst case time complexity.
   An array with 1000 elements
   An ordered array (increased or decreased) with 1000 elements

6. Sort the contents of a text file using insertion sort & hence find the time required to sort the contents.

7. Obtain the topological ordering of a vertices in a given digraph using DFS.

8. a. Implement 0/1 Knapsack problem using Dynamic Programming.
   b. Find the Binomial Co-efficient using Dynamic Programming.

9. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkistra`s algorithm.

10. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal`s Algorithm.

11. Print all the nodes reachable from a given starting node in a digraph using Breadth First Search method.

12. Sort a given set of integer elements using Heap Sort and hence find the time complexity.

13. Implement Horspool Algorithm for string matching.

14. Implement all pair shortest path problem using Floyd`s algorithm.

Implements N Queen`s problem using Backtacking.

UIS409L: OBJECT ORIENTED PROGRAMMING LABORATORY
1.5 CREDITS(0-0-3)

1. **Program using array of class objects**
   Create a class STUDENT with data members rollno, name and percentage. Create 'n' number of objects of class STUDENT and initialize the data members of each object and display them. Find the ranker among the 'n' students.

2. **Program using constructors and destructors**
   Write a C++ program to create an object of class BOOK with title of the book, author, publisher and price as its data members. Write member functions to search for title of the book and display the data. Illustrate the constructor and destructor functions associated with the BOOK object.

3. **Program using inheritance**
   Write a C++ program to illustrate multilevel inheritance.

4. **Program using inheritance**
   Write a C++ program to illustrate hybrid inheritance using hierarchical-multiple inheritance.

5. **Program using function overloading**
   Write a C++ program to overload a member function search () to search an integer key value and a key value of type double.

6. **Program using Operator overloading**
   Write a C++ program to define a class called COMPLEX containing two members real and imaginary and illustrate how the binary operators +,-,* and / can be overloaded for the objects of type COMPLEX.

7. **Program using Operator overloading**
   Write a C++ program to define a class called STACK and implement PUSH, POP and DISPLAY operations using overloaded increment and decrement operators.

8. **Program using Operator overloading**
   Write a C++ program to define a class called STRING containing two strings
   i. Overload operator + for concatenating the two strings
   ii. Overload new and delete operators for allocating and deallocating memory for
   string objects

9. **Program using I/O Streams**
   Write a C++ program to create a file called Source.txt and print its contents to the file Destin.txt so that the contents in Destin.txt are reverse of Source.txt.

10. **Program using I/O Streams**

Write a C++ program to create a text file and determine whether the end of file condition is encountered or not.

11. Program using function templates

Write a C++ program to create a template function for quick sort and demonstrate sorting of integers and double data type numbers.

12. Program using class templates

Write a C++ program to create a template class QUEUE with add, delete and display member functions. Using this class template, implement a queue of integers and doubles.

13. Program on linked list

Write a C++ program to create a class LIST (singly linked list) with member functions.

i. Insert a node at
   a. The front of a list
   b. The back of the list
   c. Any specified position in the list
ii. Delete a node from a specified position
iii. Search a node and display its position.

Demonstrate the operations by displaying the contents of the list after every operation.

14. Program on binary tree

Write a C++ program to create a class called BINTREE (Binary Tree) with member functions to perform in-order, preorder and postorder traversals. Create a BIN_TREE object and demonstrate the traversals.

UMA400M: ADVANCED MATHEMATICS-II
(MANDATORY SUBJECT)
(COMMON TO ALL BRANCHES)

1) Solid Geometry:
Distance formula (without proof), Division formula, Direction cosines and Direction ratios, planes and straight lines, angle between the planes **12 Hours**

2) Vectors:
Vector Algebra:
Vector addition, multiplication (Dot and Cross products), Triple products. Vector differentiation: Velocity , Acceleration of a vector point function , gradient, curl and divergence, solenoid and irrotational fields, simple and direct problems
12 Hours

3) Laplace transforms:
Definitions, Laplace transforms of elementary functions, derivatives and integrals. Inverse transforms, Applications. Laplace transforms to differential equations. **16 Hours**

Text Books:

1) Higher Engineering Mathematics – B S Grewal 36th edition, Khanna Publisher
2) Higher Engineering Mathematics – H K Dass 25th Edition, S.Chand and sons publications

Question Paper Pattern for SEE (for every theory subject):
Total of eight questions, two questions from each unit carrying twenty marks each, are to be set from each unit for SEE examination. Each FULL question should have a maximum of four sub questions. A student is expected to answer FIVE full questions choosing at least one FULL question from each unit.

Evaluation scheme for theory subjects:
A student will be evaluated in the subject through Continuous Internal Evaluation (CIE) and Semester End Examinations (SEE).

The CIE is for 50 marks and consists of 3 tests of 15 marks each and an assignment for 5 marks. The CIE tests are conducted for 1 hr for 30 marks, and marks obtained are scaled down to 15.

The SEE for 50 marks is conducted as 3 hrs exam for 100 marks, then the marks obtained are scaled down to 50.

Evaluation scheme for laboratory subjects:
A student will be evaluated in the subject through Continuous Internal Evaluation (CIE) and Semester End Examinations (SEE).

The CIE is for 50 marks. The students are evaluated for 30 marks for the conduct of the laboratory assignments and journal writing. The student is evaluated trough an internal lab test for the remaining 20 marks.

The SEE for 50 marks is through a lab exam of 3 hrs duration. In the lab exam student is evaluated as following:

a. 25% of total marks for initial write-up and overall submission:	12.5 marks.
b. 50% of total marks for conduct/completion of the task:	25 marks.
c. 25% of total marks for viva-voce on the lab/subject:	12.5 marks.

The student is awarded ‘S’ to ‘F’ grade based on his/her performance in CIE and SEE taken together.