SECOND YEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER –III

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.1	Applied Mathematics-III	3	1	--	3	100	25	--	--	--	**125**
COMP 3.2	Data Structures and Algorithms-I	3	1	2	3	100	25	--	25	--	**150**
COMP 3.3	Economics and Organizational Behaviour	3	--	--	3	100	25	--	--	--	**125**
COMP 3.4	Object-Oriented Programming using C++	3	1	2	3	100	25	--	25	--	**150**
COMP 3.5	Logic Design	3	1	2	3	100	25	--	--	25	**150**
COMP 3.6	Software Engineering	3	1	2	3	100	25	25	--	--	**150**
TOTAL		**18**	**05**	**08**	**--**	**600**	**150**	**25**	**50**	**25**	**850**

A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

SECOND YEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER –IV

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.1	Discrete Mathematics	3	1	--	3	100	25	--	--	--	**125**
COMP 4.2	Computer Organization	3	1	2	3	100	25	25	--	--	**150**
COMP 4.3	Microprocessors and Interfacing	3	1	2	3	100	25	--	25	--	**150**
COMP 4.4	Data Structures and Algorithms-II	3	1	2	3	100	25	--	--	25	**150**
COMP 4.5	Signals and Systems	3	1	--	3	100	25	--	--	--	**125**
COMP 4.6	Java Programming	3	1	2	3	100	25	--	25	--	**150**
TOTAL		**18**	**06**	**08**	**--**	**600**	**150**	**25**	**50**	**25**	**850**

A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

THIRD YEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER –V

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 5.1	Data Communication	3	1	--	3	100	25	--	--	--	**125**
COMP 5.2	Automata Languages and Computation	3	1	2	3	100	25	25	--	--	**150**
COMP 5.3	Coding Theory	3	1	--	3	100	25	--	--	--	**125**
COMP 5.4	Computer Hardware Design	3	1	2	3	100	25	--	--	25	**150**
COMP 5.5	Database Management System	3	1	2	3	100	25	--	25	--	**150**
COMP 5.6	Operating Systems	3	1	2	3	100	25	--	25	--	**150**
TOTAL		**18**	**06**	**08**	**--**	**600**	**150**	**25**	**50**	**25**	**850**

A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

THIRD YEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER –VI

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 6.1	Design Patterns and Frameworks	4	--	--	3	100	25	--	--	--	**125**
COMP 6.2	Design and Analysis of Algorithms	3	1	2	3	100	25	25	--	--	**150**
COMP 6.3	Artificial Intelligence	3	1	2	3	100	25	--	25	--	**150**
COMP 6.4	Computer Graphics	3	1	2	3	100	25	--	25	--	**150**
COMP 6.5	Device Interface and PC Maintenance	3	1	--	3	100	25	--	--	--	**125**
COMP 6.6	Network Programming	3	1	2	3	100	25	--	--	25	**150**
TOTAL		**19**	**05**	**08**	**--**	**600**	**150**	**25**	**50**	**25**	**850**

A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

FINAL YEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER –VII

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 7.1	Compiler Construction	3	1	2	3	100	25	--	--	25	**150**
COMP 7.2	Data Mining	3	1	2	3	100	25	--	--	--	**125**
COMP 7.3	Image Processing	3	1	2	3	100	25	--	25	--	**150**
COMP 7.4	Elective-I	3	1	2	3	100	25	--	--	--	**125**
COMP 7.5	Elective-II	3	1	--	3	100	25	--	--	25	**150**
COMP 7.6	Project	--	--	4	--	--	--	--	--	25	**25**
TOTAL		**15**	**5**	**12**	**--**	**500**	**125**	**--**	**25**	**75**	**725**

A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

List of Electives

Subject Code	Elective-I	Subject Code	Elective-II
COMP 7.4.1	VLSI Design	COMP 7.5.1	Big Data Analytics
COMP 7.4.2	Data Compression	COMP 7.5.2	Geographical Information Systems
COMP 7.4.3	Fuzzy Logic and Neural Networks	COMP 7.5.3	Bio Informatics
COMP 7.4.4	Digital Signal Processing	COMP 7.5.4	Project Management and Quality Assurance
COMP 7.4.5	Cloud Computing	COMP 7.5.5	Entrepreneurship Development

FINAL YEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER –VIII

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 8.1	Web Technology	3	1	2	3	100	25	--	25	--	**150**
COMP 8.2	Cyber Security	3	1	2	3	100	25	--	--	--	**125**
COMP 8.3	Elective-III	3	1	2	3	100	25	--		25	**150**
COMP 8.4	Elective-IV	3	1	2	3	100	25	--	--	25	**150**
COMP 8.5	Project*	--	--	8	--	--	--	75	--	75	**150**
TOTAL		**12**	**04**	**16**	**--**	**400**	**100**	**75**	**25**	**125**	**725**

* Term Work in Project is a separate Head of Passing.

A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

List of Electives

Subject Code	Elective-III	Subject Code	Elective-IV
COMP 8.3.1	Embedded System Design	COMP 8.4.1	Genetic Algorithms
COMP 8.3.2	Multimedia Systems	COMP 8.4.2	Statistical Models for Computer Science
COMP 8.3.3	Distributed Operating Systems	COMP 8.4.3	Mobile Computing
COMP 8.3.4	Pattern Recognition	COMP 8.4.4	Functional Programming
COMP 8.3.5	Web Engineering	COMP 8.4.5	Natural Language Processing

FE 1.5 FUNDAMENTALS OF COMPUTER ENGINEERING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
FE 1.5	Fundamentals of Computer Engineering	3	--	2	3	100	25	--	--	--	**125**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of basic concepts of computer science and engineering.
2. An introduction to the fundamentals of hardware, software and programming.
3. An introduction to mathematical software.
4. An understanding of cyber laws and computer security.

Course Outcomes:

The student after undergoing this course will be able to:

1. Demonstrate the use of mathematical software and solve simple mathematical problems.
2. Explain the needs of hardware and software required for a computation task.
3. State typical provisions of cyber law that govern the proper usage of Internet and computing resources.
4. Explain the working of important application software and their use to perform any engineering activity.
5. Demonstrate the use of Operating system commands and shell script.

UNIT -1 (12 hours)

Overview, Introduction to computers: Generation of Computers. Software and hardware, Types of Computers, Computer Networks and Internet. Data and program representation. Working of CPU, Making computers faster and better now and in the future. Storage systems characteristics, types of storage systems, Magnetic disk systems, Optical disk systems and Flash Memory systems. Keyboards, Pointing devices, Scanners, Readers and Digital cameras, Audio input, Display devices, Printers, Audio output.

UNIT- 2 (12 hours)

Introduction to System software and Application software, the operating system (OS). OS for Desktop PCs, servers, handheld PCs, Smartphone and larger

computers. Linux and Windows Operating system commands and shell scripts. Concepts of Word processing, Spreadsheet, Database, Presentation graphics and multimedia. Introduction to Assemblers, Interpreters, Compilers and Debuggers.

UNIT-3 **(12 hours)**

Basic Concepts of Technology and Law, Understanding the Technology of Internet, Scope of Cyber Laws, Cyber Jurisprudence, Encryption, Science of Cryptography, Symmetric and Asymmetric Cryptography. Electronic Banking: Banking and Bookkeeping, Legal Recognition of Digital Signature. The Cyber Crime, Tampering with Computer Source Document, Hacking with Computer System.

UNIT-4 **(12 hours)**

MATLAB and Its family, Menus and toolbars, Types of windows and types of files, MATLAB Help system, Basic calculations in MATLAB, Vectors and arrays, Multi-dimensional arrays, Element by element operations, Polynomial operations using arrays, X-Y Plotting functions, Subplots, 3-D Plots and Contour plots.

Recommended Readings:

1. Deborah Morley and Charles S. Parker; Fundamentals of Computers; Cengage Learning, India edition; 2009.
2. Alexis Leon and Mathews Leon; Fundamentals of Information Technology; Vikas Publication, Chennai.
3. Francis Scheid; Theory and Problems of Introduction to Computer Science Schaum's Outline Series; Tata McGraw Hill publication.
4. Information Technology: Tools and Application, Ed. UPTEC Computer Consultancy Limited, Elsevier Publication, 2004.
5. Rudra Pratap ;Getting started with MATLAB: A quick introduction for scientists and engineers; Oxford University press; 2003.
6. W. L. Palm III ; Introduction to MATLAB 7 for Engineers; McGraw Hill; 2005.
7. Rajeshree R Khande and Manisha Maddel ; Internet Programming & Industrial Law; Vision Publications, Pune.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1) Five programs using MATLAB (Programs will be on Basic Calculation, Calling Data file and Sending results to Data file, Control structure, Plots and Subplots, creating and using built in functions)
2) Five programs using linux shell scripting. (Using any scripting language like PERL or PYTHON)
3) Five experiments involving packages for Word Processing, Spread Sheet, Presentation, Graphics and Database.

FE 2.3 PROGRAMMING LANGUAGES

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
FE 2.3	Programming Languages	3	--	2	3	100	25	--	--	--	**125**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of basic concepts of computer programming and developer tools.
2. An introduction to the syntax and semantics of the "C" language as well as data types offered by the language.
3. An introduction to write programs using standard language infrastructure regardless of the hardware or software platform.

Course Outcomes:

The student after undergoing this course will be able to:

1. Demonstrate the use of algorithms and flowcharts to plan the solution of a computing problem.
2. Explain the use of formatted and unformatted input and output statements in "C".
3. State typical usage of sequence control statements of "C'.
4. Enlist the fundamental data types and data structures of "C".
5. Explain the usage of arrays and pointers in "C".
6. Differentiate between a structure and a union.
7. Explain the commands of File Management in "C".

UNIT - 1 (12 Hours)

Programming Basics: Notions of algorithms, flowcharts and programming, iteration and recursion. Imperative style of programming, Functional style of programming, correctness and efficiency issues. Features of block-structured languages, Functions and procedures, Parameter passing, Top-down style and stepwise-refinement with concrete examples Fundamental algorithms: Exchanging values of two variables, counting, summation of a set of numbers , generation of prime numbers , reversal ,series.

UNIT - 2 (12 Hours)

Overview of Programming language C, constants variables and data types, operators and expressions, data input output, decision making and looping: If, If-else, while, do-while, for, switch. Function declarations and prototypes, pass by value, and pass by reference. User defined function in C, iterative function and recursive functions.

UNIT - 3 (12 Hours)

Arrays: One dimension array, array initialization, Searching, Insertion, deletion of an element from an array; finding the largest/smallest element in an array, two dimension array, addition/multiplication of two matrices, transpose of a square matrix; passing array to function , character array and string. Pointers: Address operators, pointer type declaration, pointer assignment, pointer initialization, pointer arithmetic, functions and pointers, arrays and pointers, pointer arrays.

UNIT - 4 (12 Hours)

Structure & Unions**:** Defining a structure, declaring structure variables, Accessing structure members, structure initialization, copying & comparing structure variables, operation on individual members, Array of structures, structure & functions, Unions, Size of Structure.
Files management in C: Defining & opening a file, closing a file, I/O operations on files, Error handling during I/O files, Random Access to files. Introduction to Dynamic Memory Allocation

Recommended Readings:

1. Herbert Schildt ; C: The Complete Reference, 4th Edition; Tata McGraw Hill; 2000.
2. Stephen Prata ; C Primer Plus 5th Edition; SAMS Publishing; 2005.
3. Brian W. Kernighan and Dennis M. Ritchi; C Programming Language 2nd Edition; Pearson Education; 2006.
4. Samuel P. Harbison and Guy L. Steele; C: A Reference Manual , 5th Edition; Prentice Hall; 2003.
5. Yashwant Kanetkar; Let Us C; BPB Publications, 9th Edition; 2008.
6. King K.N; C Programming: A Modern Approach, 2nd Edition; W. W. Norton and Company; 2008.
7. Dromey R.J ; How to Solve it by Computer, Prentice Hall India Series; 2000.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Program to find area and circumference of circle.
2. Program to convert temperature from degree centigrade to Fahrenheit.
3. Program to calculate sum of 5 subjects & find percentage.
4. Program to show swap of two no's without using third variable.
5. Program to print a table of any number.
6. Program to find greatest in 3 numbers.
7. Program to show the use of conditional operator.
8. Program to find whether given no is even or odd.
9. Program to shift inputed data by two bits to the left.
10. Program to use switch statement. Display Monday to Sunday.
11. Program to display first 10 natural no & their sum.
12. Program to print Fibonacci series up to 100.
13. Program to find factorial of a number.
14. Program to find whether given no is a prime no or not.
15. Program to display series and find sum of 1+3+5+……..+n.
16. Program to use bitwise AND operator between the two integers.
17. Program to add two number using pointer.
18. Program to show sum of 10 elements of array & show the average.
19. Program to find sum of two matrices.
20. Program to find multiplication of two matrices.
21. Program to find transpose of a matrix.
22. Program to find the maximum number in array using pointer.
23. Program to reverse a number using pointer.
24. Program to show input and output of a string.
25. Program to find square of a number using functions.
26. Program to show call by value.
27. Program to show call by reference.
28. Program to find factorial of a number using recursion.
29. Program to find whether a string is palindrome or not.

COMP 3.1 Applied Mathematics-III

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.1	Applied Mathematics-III	3	1	--	3	100	25	--	-	--	**125**

Course Objectives: The aim of learning this course is to provide students with the mathematical knowledge and skills necessary to support their concurrent and subsequent engineering studies.

Course Outcomes:

After successful completion of this course the student would be able to

1. Compute the rank and inverse of a matrix and solve system of linear equations.
2. Compute Eigen values and Eigen vectors of a given matrix, apply Cayley Hamilton theorem.
3. Understand the basic concepts of probability, random variables, mean, variance, standard deviation and probability distributions, correlation and regression.
4. Use tools like Laplace transforms and Fourier transforms in formulating and solving Engineering problems.
5. Understand Z- transforms and its properties and apply it in solving difference equations.

UNIT - 1 (12 Hours)

Linear Algebra: Types of matrices, adjoint, inverse. Elementary transformations.Rank of a matrix, normal form, echelon form. Linear system of equations AX = B and AX = 0. Linearly independent and dependent vectors, Eigen values and Eigen vectors, Cayley Hamilton Theorem, minimal equation, Diagonalization.

UNIT - 2 (14 Hours)

Probability and Probability distributions: Definition, properties, Axioms of probability, Conditional probability, Baye's theorem, Random Variables. Discrete probability distribution, Continuous probability distribution, Distribution function. Expectation and Variance, Moment generating function. Special distributions: Binomial, Poisson, Geometric, Normal, Uniform and exponential. Correlation and regression.

UNIT - 3 (10 Hours)

Laplace Transforms: Definition, Existence conditions, properties, inverse Laplace transforms. Laplace transform of periodic functions, Convolution theorem, Laplace transform of Dirac-Delta function, Application of Laplace transforms in solving linear differential equations with initial conditions and system of linear simultaneous differential equations.

UNIT - 4 (12 Hours)

Fourier and Z-transforms: Definition, properties, inverse. Convolution theorem. Applications of Fourier and Z-transforms.

Recommended Readings:

1. Grewal B. S.; Higher Engineering Mathematics; Khanna Publications, New Delhi.
2. H. K. Dass; Advanced Engineering Mathematics; S. Chand &Co.
3. Erwin Kreyzing; Advanced Engineering Mathematic; Wiley.
4. Kandasamy, P.; Engineering Mathematics; Chand & Co., New Delhi.
5. Srimanta Pal, Subodh C. Bhunia; Engineering Mathematics; Oxford University Press.
6. Dr. D. S. C ; Engineering Mathematics- Part III ; Prism Books Pvt. Ltd.
7. Montgomery, D. C., Probability and Statistics for Engineers; Prentice Hall of India.

COMP 3.2 DATA STRUCTURES AND ALGORITHMS-I

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.2	Data Structures and Algorithms-I	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An ability to use data structures as the foundational base for computer solutions to engineering problems.
2. An understanding of the different logical relationships among various data items.
3. Ability to understand the generic principles of computer programming as applied to sophisticated data structures.
4. An ability to plan, design, execute and document sophisticated technical programs to handle various sorts of data structures.

Course Outcomes:

The student after undergoing this course will be able to:

1. Design algorithms using principles of recursion.
2. Demonstrate the use of data structures like linked lists , stacks and queues.
3. Explain the applications of linked lists, stacks and queues in Computer Engineering.
4. Apply the knowledge of data structures to a given problem.
5. Illustrate searching, sorting and hashing techniques.

UNIT -1 (12 Hours)

Defining, Declaring and Initialization of Structure variables. Accessing members of a structure, Array of structures, Nested Structures, Pointers to structures. Passing structure, structure members, structure arrays and pointer to structure as function parameters. Self referential structures. Introduction to Data Structures: Linear and Non Linear Data Structures, Static and Dynamic Data Structures. Array Implementation of LIFO and FIFO data structures: Stack and Queue.

UNIT -2 (12 Hours)

Concept of Linked Lists. Singly linked lists, Doubly linked lists and circular linked lists. Insertion, deletion, update and copying operations with Singly linked

lists, Doubly linked lists and Circular linked lists. Reversing a singly linked list. Circular Doubly Linked List: Linked list with Header Node, Sorted Linked List, Merging, Concatenation Comparison of Arrays and linked Lists. Linked List Implementation of Stack, Linked List implementation of Queue, Circular Queue, Deque and Priority Queue.

UNIT -3 (12 Hours)

Recursion: Writing a recursive function, Flow of control in recursive functions, Winding and unwinding phase, Recursive data structures, Implementation of recursion. Tail recursion. Indirect and Direct Recursion. Applications of Linked Lists**:** Polynomial arithmetic with linked list, Creation of polynomial linked list, Addition of 2 Polynomials, Multiplication of 2 polynomials. Applications of Stacks**:** Reversal of a String, Checking validity of an expression containing nested parenthesis, Function calls, Polish Notation: Introduction to infix, prefix and postfix expressions and their evaluation and conversions. Application of Queues: Scheduling, Round Robin Scheduling.

UNIT -4 (12 Hours)

Searching: Sequential Search, Binary Search. Hashing: Hash Functions: Truncation, Midsquare Method, Folding Method, Division Method. Collision Resolution: Open Addressing: Linear Probing, Quadratic Probing, Double Hashing, Deletion in open addressed table, implementation of open addressed tables, Separate Chaining Bucket Hashing. Sorting: Sort order, Types of sorting, Sort stability, Sort by address, In place sort, Sort pass, Sort efficiency. Selection sort, Bubble sort, Insertion sort, Shell sort, Merge sort, Quick sort, Radix sort and Address Calculation Sort.

Recommended Readings:

1. S. K Srivastava, Deepali Srivastava; Data Structures through C in Depth; BPB Publications; 2011.
2. Yedidya Langsam, Moshej Augenstein, Aaron M. Tenenbaum; Data Structure Using C & C++; Prentice Hall of India; 1996.
3. Rajesh K. Shukla; Data Structures using C and C++; Wiley India; 2009.
4. Ellis Horowitz, Sartaj Sahni; Fundamentals of Data Structures; Galgotia Publications; 2010.
5. Jean Paul Tremblay, Paul G. Sorenson; An introduction to data structures with applications; Tata McGrawHill; 1984.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Implementation of array of structures.
2. Implementation of pointers to structures.
3. Implementation of stack using array.
4. Implementation of queue using array.
5. Implementation of singly linked list.
6. Implementation of doubly linked list.
7. Implementation of circular linked list.
8. Implementation of stack using linked list.
9. Implementation of queue using linked list.
10. Implementation of conversion of infix to postfix and evaluation of postfix.
11. Implementation of Tower of Hanoi using recursion.
12. Implementation of sequential search in an array.
13. Implementation of binary search in an array.

COMP 3.3 ECONOMICS AND ORGANIZATIONAL BEHAVIOUR

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.3	Economics and Organizational Behaviour	3	--	--	3	100	25	--	--	--	**125**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of demand and supply.
2. An understanding of Financial management
3. An understanding of the role of Communication Function in organizations.
4. An understanding of the complexity of managing in a global world.
5. An understanding of the concepts of Organizational Behaviour.

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain economics using demand and supply.
2. Apply the concepts of Financial Engineering
3. Explain the role of Communication Function in organizations.
4. Apply managerial concepts to solve complex problems related to global issues.
5. Explain the essential components of an organization.
6. Explain the essential requirements to become a successful entrepreneur.

UNIT -1 **(12 Hours)**

Introduction to Economics and general concepts: Demand and Supply- Demand curve, Supply curve, Market Equilibrium. National Income terms: GDP, Real v/s Nominal GDP, Net Domestic Product, GNP, National Income, Per capita income, Disposable Income, Price Index, Inflation. Estimation/Forecasting of Demand- Meaning, importance, methods–trend, exponential smoothing, regression analysis. Economies and diseconomies of Scale.

UNIT -2 **(12 Hours)**

Financial Management: An Overview – Evolution of financial management, Financial decisions in a firm, The Fundamental principle of Finance, business ethics and social responsibility. Time value of money – times lines and notations, future value of a single amount, present value of a single amount, future value of

an annuity, present value of an annuity. Capital Budgeting - Different Methods of Evaluation of Projects- Payback Period, Discounted Cash Flow methods- Net Present Value, Internal Rate of Return. Leasing – types of leases, Rationale of leasing, mechanics of leasing.

UNIT -3 (12 Hours)

Working Capital Management: Determinants of working capital, financing of working capital, dangers of excessive and shortage of working capital. Equity capital, preference capital and term loans. Preparation of Income statement, Balance sheet, Fund Flow statement. Understanding and analyzing them using financial ratios – liquidity, leverage and profitability ratios. Mergers, Takeovers and Acquisitions.

UNIT -4 (12 Hours)

Understanding of Organizational Behavior Nature of Organizations. Nature and Importance of Communication. The Two-Way Communication Process, Communication Barriers, Communication Symbols, Downward and Upward Communication, Formal and Informal Communication. Forms of Communication. Model of Motivation. Motivational Drives. Human Needs, Types of Needs. Maslow's Hierarchy of Needs. Hezberg's Two-Factor Theory. Behavior Modification. Goal Setting, Motivational Applications. The Expectancy Model. Nature of Work Change. Three Stages in Change. Reaching a New Equilibrium. The Organizational Learning Curve for Change. Appraising and Rewarding Performance. Organizational Behavior and Performance Appraisal. Economic Incentives Systems. The Reward Pyramid.

Recommended Readings:

1. R. L. Varshney and K L Maheswari; Managerial Economics; Nineteenth, Revised and Enlarged Edition; Sultan Chand and Sons Publications.
2. Peterson, Lewis; Managerial Economics; P.H.I.
3. Prasanna Chandra; Fundamentals of Financial Management, Third Edition;Tata McGraw-Hill, New Delhi.
4. Richard M. Lynch and Robert W. Williamson; Accounting for Management, Planning and Control, Third Edition; Tata McGraw-Hill, New Delhi.
5. John W. Newstrom and Keith Davis; Organizational Behavior (Human Behavior at Work), Tenth Edition; Tata McGraw Hill.

COMP 3.4 OBJECT ORIENTED PROGRAMMING USING C++

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.4	Object Oriented Programming using C++	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of the concept of object oriented programming.
2. An understanding of the concepts of data hiding, data abstraction, polymorphism inheritance and exception handling.
3. Ability to understand the generic principles of object oriented programming using "C++".
4. An understanding the use of templates in "C++".
5. An ability to plan, design, execute and document sophisticated object oriented programs to handle different computing problems.

Course Outcomes:

The student after undergoing this course will be able to:

1. Design algorithms using principles of object oriented programming.
2. Demonstrate the concepts of data abstraction and data hiding using 'C++".
3. Explain the applications of polymorphism and inheritance in object oriented programming.
4. Apply the knowledge of standard template library achieve reusability.
5. Illustrate stream I/O and exception handling.

UNIT -1 (12 Hours)

Introduction to Classes and Objects, Functions and an Introduction to Recursion, Arrays and Vectors, Pointers.

UNIT -2 (12 Hours)

Class scope and accessing class members, Constructors and destructors, Const objects and const member functions, Friend functions and friend classes, This pointer, Static class member, Data Abstraction and information hiding, Operator overloading, Inheritance, Polymorphism.

UNIT -3 (12 Hours)

Templates, Stream input/ output, Exception handling, File processing.

UNIT -4 (12 Hours)

String processing. Standard Template Library: Introduction to standard template library. Sequence Containers: vector, list, deque. Associative Containers: Set, Multiset, Map. Container Adapter: stack, queue, priority queue adapter
Algorithms: fill, fill_in, generate, generate_n, Lexicographical compare, Replace, replace_if

Recommended Readings:

1. Paul Deitel and Harrey Dietel; C++, How to Program; seventh edition.
2. Stanley Lippman; C++ Primer; Fifth edition.
3. Herbert Schildt; Complete Reference; Fourth edition.
4. Bjarne Stroustrup; C++ Programming Language; Fourth edition.
5. D Ravichandran; Programming with C++; Third Edition.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Classes and objects.
2. Friend function and friend classes.
3. Function overloading.
4. Operator Overloading.
5. Constructors and Destructors.
6. Inheritance.
7. Polymorphism and virtual functions.
8. Stream Input Output.
9. Exception Handling.
10. Templates.
11. File Handling.
12. Standard Template Library.

COMP 3.5 LOGIC DESIGN

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 3.5	Logic Design	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of various Number Systems & Codes along with Boolean algebra.
2. An ability to solve Boolean algebra problems.
3. An ability to design combinational and sequential circuits.
4. An understanding of programmable logic devices.

Course Outcomes:

The student after undergoing this course will be able to:

1. Convert the numbers from one radix to another and perform arithmetic operations using the 1's and 2's compliments.
2. Solve Boolean Expressions using Boolean algebra, K-maps and VEM and implement them using logic gates.
3. Design any given combinational circuits.
4. Explain different flip flops, registers and their applications.
5. Design sequential circuits and state machines.
6. Design synchronous and asynchronous counter circuits.
7. Explain arithmetic circuits like adders and multipliers and their applications.
8. Compare the characteristics of programmable logic devices.

UNIT -1 (12 Hours)

Introduction: Digital and analog systems, Logic levels and Pulse Waveforms.
Number systems – Decimal, Binary, Representation of Signed numbers and binary arithmetic, Octal number system, Hexadecimal number system.
Binary codes – Classification, 8421 BCD code, XS-3 code, Gray code, Error correction and detection codes. Logic gates-AND, OR, NOT, Universal , X-OR, X-NOR gates.
Boolean algebra: Logic operations, Laws of Boolean Algebra, Duality, Reducing Boolean expressions, Boolean functions and their representations, Boolean

expressions in SOP and POS forms, Computation of total gate inputs, Boolean expressions and logic diagrams, Conversion of AOI to NAND / NOR logic.

UNIT -2 (12 Hours)

Minimization of Switching Functions: Two, Three, Four variable K-Map, Don't Care Combinations, Quine-McCluskey method.
Combinational logic Design: Adders, Subtractors, Binary Parallel Adder/ Subtractor, Look Ahead Carry Adder, Code Converters, Parity generators/checkers , Comparators, Encoders, Decoders, Multiplexers and De-multiplexers, Modular design using IC chips.

Programmable logic devices: ROM, ROM Organization, Combinational Circuit implementation, Types of ROM, Combinational Programmable logic devices, PAL, PLA, PROM, Programmable logic devices Comparison.

UNIT -3 (12 Hours)

Flip-flops: Classification of Sequential Circuits, Latches & flip-flops - D flip-flop, JK flip-flop, T flip-flop. Flip-flop operating characteristics, Race around condition, Master slave flip-flop, conversion of one flip-flop to another , Applications of flip-flop.
Shift Registers: Buffer register, Data Transmission in Shift Registers, Serial-In Serial-Out Shift register, Serial-In Parallel-Out Shift register, Parallel-In Serial-Out Shift register, Parallel-In Parallel-Out Shift register, Bidirectional shift register, Universal Shift register, Applications of Shift register.

UNIT - 4 (12 Hours)

Counters: Asynchronous counters, Design of asynchronous counters, Synchronous counters, Shift register counters.
Sequential Circuits : Finite state model, Memory elements, Synthesis of synchronous sequential circuits, Serial Binary Adders, Sequence Detector.

Recommended Readings:

1. A. Anand Kumar; Fundamentals of Digital circuits; PHI, Second Edition.
2. Thomas L. Floyd; Digital Fundamentals; Prentice Hall.
3. Morris Mano; Digital Logic and Computer Design; PHI Publication.
4. Malvino & Leach; Digital Principles and Applications; TMH Publication.
5. R. P. Jain; Modern Digital Electronics; TMH Publication.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Performance of Basic Logic Gates.
2. a. De Morgan's Theorem (first and second law).
   b. Associative, Cumulative and Distributive laws.
3. Universal Gates-NAND and NOR .
4. Binary to Gray code conversion.
5. Half Adder and Full Adder.
6. Half Substractor and Full Substractor.
7. Sum of Product.
8. BCD to XS-3.
9. BCD to Seven-Segment Display.
10. Flip-Flop.
    a. SR-Flip Flop and D-Flip Flop.
    b. JK-Flip Flop and T-Flip Flop.

COMP 3.6 SOFTWARE ENGINEERING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
					Th Duration (Hrs)	Marks					
		L	T	P		Th	S	TW	P	O	Total
COMP 3.6	Software Engineering	3	1	2	3	100	25	25	--	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of the current issues and practices in software engineering with an emphasis on the software development process.
2. An ability to understand the software planning and management.
3. Ability to plan software requirements specifications, system modeling, quality specifications, and program specifications.
4. An understanding of software design approaches.
5. An understanding of the requirements of software project management.
6. An ability to recognize social, ethical, cultural, and safety issues in software deployment.

Course Outcomes:

The student after undergoing this course will be able to:

1. Design a specification a software system for any existing system.
2. Plan a design of software system as per the specification.
3. Implement a software system it with readable, reusable, modular and object-oriented techniques.
4. Design a test procedure for validity, correctness and completeness.
5. Implement a software maintenance schedule.
6. Demonstrate the skills of a Software Designer, Software Architect or Project Manager for the development of software to solve business and technical problems.
7. Explain the methodologies, architectural approaches, project management techniques, and team dynamics.

UNIT -1 **(12 Hours)**

Scope of software engineering: Historical Aspects, Economic Aspects, Maintenance Aspects, Requirements, Analysis and Design Aspects, Team Development Aspects. Software Life-Cycle Models: Code-and-Fix Life-Cycle Model, Waterfall Life-Cycle Model, Rapid-Prototyping Life-Cycle Model, Open Source Life-Cycle Model, Agile Processes, Synchronize-and-Stabilize Life-Cycle Model, Spiral Life-Cycle Model. Software Process: The Requirements Workflow, The Analysis Workflow, The Design Workflow, The Implementation Workflow,

The Test Workflow, Post-delivery Maintenance, Retirement Capability Maturity Models Teams: Team Organization, Democratic Team Approach, Classical Chief Programmer Team Approach, Synchronize-and-Stabilize Teams, Teams for Agile Processes, Open-source Programming Teams.

UNIT -2 (12 Hours)

The Tools of the Trade: CASE, Taxonomy of CASE, Scope of CASE, Software Versions, Configuration Control. From Modules to Objects: Cohesion, Coupling, Data Encapsulation – Data Encapsulation and Development, The Object-Oriented Paradigm.

Reusability and Portability: Reuse Concepts, Impediments to Reuse, Objects and Reuse, Reuse during Design and Implementation – Design Reuse, Portability. Techniques for Achieving Portability. More on UML: Class Diagrams, Use-Case Diagrams, Interaction Diagrams, State Charts, Activity Diagrams.

UNIT -3 (12 Hours)

Testing: Quality Issues, Non-Execution-Based Testing, Execution-Based Testing, Testing versus Correctness Proof and stopping criteria. Planning and Estimating: Planning and the Software Process, Estimating Duration and Cost. Requirements: Determining what the Client Needs, Overview of the Requirements Workflow, Understanding the domain, The Business Model, Initial Requirements, Metrics and Challenges for Requirement Workflow.Classical Analysis: The Specification Document, Informal Specifications, Structured Systems Analysis, Metrics and Challenges for Classical Analysis.

UNIT -4 (12 Hours)

Design and Abstraction, Operation Oriented Design, Data Flow Analysis, Data-Oriented Design, Object-Oriented Design, Challenges and Metrics for Design Workflow Implementation: Choice of Programming Language, Fourth-Generation Language, Good Programming Practice, Coding Standards, Code Reuse, Integration, Test Case Selection, Black-Box Unit-Testing Techniques, Glass-Box Unit-Testing Techniques, Code Walkthroughs and Inspections, Integration Testing, Product Testing, Acceptance Testing. Post delivery Maintenance: Development and Maintenance, Management of Post delivery Maintenance, Maintenance of Object Oriented Software, Post delivery Maintenance Skills versus Development Skills, Reverse Engineering, Testing during Post delivery Maintenance, Metrics and Challenges for Post delivery Maintenance.

Recommended Readings:

1. Stephen R. Schach; Software Engineering; TMH, Seventh Edition.
2. Roger S. Pressman; Software Engineering – A practitioner's approach; McGraw Hill, Seventh Edition.
3. Edward Kit; Software Testing in the Real World: Improving the Process; Addison – Wesley Publishing company; 1995.
4. Pankoj Jalote; Software Project Management in Practice; PEA
5. Ian Sommervilee; Software Engineering; 10th Edition Pearson.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. The Term Work Marks to be awarded based on the assessment of experiments conducted.)

1. Introduction to Software Crisis and Software Processes.
2. Requirements (Requirements Document).
3. Specifications (Software Requirement Specification).
4. Design.
5. Cost Estimation.
6. Implementation.
7. Black Box Testing.
8. White Box Testing.
9. Software Reliability.
10. Software Maintenance.

COMP 4.1 Discrete Mathematics

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.1	Discrete Mathematics	3	1	--	3	100	25	--	-	--	**125**

Course Objective: This course is designed to introduce students to the techniques, algorithms, and reasoning processes involved in the study of discrete mathematical structures that are essential to the field of Computer Science.

Course Outcomes: On completing this course students will be able to

1. Perform operations on discrete structures such as sets, functions, relations, and sequences.
2. Know the properties of equivalence relations and partial orderings.
3. Apply algorithms and use definitions to solve problems to prove statements in elementary number theory.
4. Construct mathematical arguments using logical connectives and quantifiers and verify the correctness of an argument using propositional and predicate logic and truth tables.
5. solve problems using the basic principles of counting theory, including permutation, combinations, and the pigeonhole principle
6. Solve problems involving recurrence relations and generating functions.
7. Understand lattices and Boolean algebras.
8. Explain basic definitions and properties associated with simple planar graphs, including isomorphism, connectivity, and Euler's formula, and describe the difference between Eulerian and Hamiltonian graphs.
9. Use graphs and trees as tools to solve combinatorial optimization problems

UNIT - 1 (12 Hours)

Set Theory : Sets, Set Operations, Relations and their properties, Equivalence Relations, partial orderings.
Functions: One-to-One and Onto Functions, Inverse Function, Composition of functions, Graphs of functions and some important functions.
Integers: Integers and division (excluding applications of congruences and cryptology), primes and greatest common divisors, Integers and algorithms.

UNIT - 2 (12 Hours)

Propositional Calculus: Propositional logic, propositional equivalences, predicates and quantifiers, rules of inference.
Boolean Algebra: Boolean functions, representing Boolean functions.
Mathematical Induction: Principle of Mathematical Induction and applications.

UNIT - 3 (12 Hours)

Counting: The basics of counting, pigeonhole principle, permutations and combinations, binomial coefficients.
Advanced Counting Techniques: Recurrence relations, solving linear recurrence relations, inclusion –exclusion principle, applications of inclusion – exclusion principle.

UNIT - 4 (12 Hours)

Graph theory: Graphs and graph models, graph terminology and special types of graphs, representing graphs and graph isomorphism, connectivity, Euler and Hamilton paths, shortest path problems, planar graphs, graph coloring.
Trees: Introduction to Trees, applications of trees, tree traversal, Spanning Trees, Minimal Spanning Trees.

Recommended Readings:

1. Kenneth H. Rosen; Discrete Mathematics and Its Applications; Tata McGraw Hill (6th edition).
2. B Kolman, R.C. Busby and Sharon C. Ross; Discrete Mathematical Structures; Prentice Hall.
3. J. P. Tremblay and R. Manohar, McGraw Hill; Discrete Mathematical Structures with Applications to Computer Science; New York McGraw Hill.
4. Swapan Kumar Sarkar; Discrete Mathematics; S.Chand Publication.
5. Dr. D. S. C ;Discrete Mathematical Structures; Prism Books Pvt. Ltd.
6. G.V.Kumbhojkar; Discrete Structures And Graph Theory; Pradeep Prakashan.

COMP 4.2 COMPUTER ORGANIZATION

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.2	Computer Organization	3	1	2	3	100	25	25	--	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of relationship between hardware and software.
2. An ability to recognize how machine organization impacts the efficiency of applications written in a high-level language.
3. An ability to understand the system performance and concepts behind advanced pipelining techniques.
4. An understanding of different ways of communicating with I/O devices and standard I/O interfaces.
5. An ability to develop solutions for basic programs using assembly language.

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the organization of the Control unit, Arithmetic and Logical unit, Memory unit and the I/O unit.
2. Identify high performance architecture design.
3. Create an assembly language program to program a microprocessor system.
4. Design a pipeline for consistent execution of instructions with minimum hazards.
5. Explain the ways to take advantage of instruction level parallelism for high performance processor design.
6. Demonstrate memory hierarchy and its impact on computer cost/performance.

UNIT - 1 (12 Hours)

Introduction to Computer Organization: Computer components, Functions, interconnection Structure, Bus Interconnection. Computer Arithmetic: Integer Representation-unsigned numbers, signed numbers, signed magnitude,2's compliment, Biased Representation. Integer Arithmetic: Addition, Subtraction, Multiplication unsigned, signed (Booths Algorithm),Division-unsigned, signed.

Floating-Point Representation: IEEE 32 bits, 64 bits. Floating-Point Arithmetic: Addition, Subtraction, Multiplication, Division.

UNIT - 2 (12 Hours)

Internal Memory: Semiconductor Memory - Memory Hierarchy, Characteristics of Memory System, Semiconductor RAM Memories, Internal Organization of Memory Chip, Static RAM, Asynchronous DRAM, Synchronous DRAM, Connection of Memory to the processor, RAM Bus memory. Cache Memory: Basics of Cache, Structure, Read operation, Elements of Cache Design. Associative Memory: External Memory: Magnetic Disk, RAID, optical Memory. Virtual Memory: Logical VS Physical Address space, working Principle, Mapping Functions, Replacement Policy.

UNIT - 3 (12 Hours)

Input/Output: External Devices, I/O Modules, Programmed I/O, Interrupt Driven I/O, Direct Memory Access, I/O Channel and Processor. CPU Structure and Functions: Processor Organization, Register Organization, Instruction Pipeline, Basic Concepts of Pipelining. RISC CPU Architecture: Instruction Execution Characteristics, Use of Large Register File, Compiler based register optimization, Reduced Instruction Set Architecture, RISC v/s CISC.

UNIT - 4 (12 Hours)

Buses: Bus interconnections, VGA, Asynchronous v/s Synchronous Buses, PCI Bus, SCSI
Control Unit Operation: Micro Operations, Control of the CPU, Hardwired Implementation Micro programmed Control: Basic Concepts, Microinstruction Sequencing, and Microinstruction Execution. Parallel Processing: Multi Processing, Cache Coherence /MESI Protocol.

Recommended Readings:

1. William Stalling; A textbook of Computer Organization and Architecture; Edition VI.
2. M. Morris Mano ; A textbook of Computer Organization and Architecture.
3. Douglas V. Hall ; Microprocessors and Interfacing.
4. David A. Patterson, John L. Hennesy ; Computer Organization And Design, Edition III.
5. Carl Hamacher, Zvonko Vranesic, Safal Zaky ; Computer Organization; Edition V.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. The Term Work Marks to be awarded based on the assessment of experiments conducted.)

1. Assembly language program to perform addition on
    a. 8 bit data.
    b. 16 bit data.
    c. 32 bit data.
2. Assembly language program to perform subtraction on
    a. 8 bit data.
    b. 16 bit data.
3. Assembly language program to perform multiplication on
    a. 8 bit data.
    b. 16 bit data.
4. Assembly language program to find average of two numbers.
5. Assembly language program to find two's compliment of number.
6. Assembly language program to check status of sign flag.
7. Assembly language program to compute Xn.
8. Assembly language program to perform bubble sort in descending order.
9. Assembly language program to find largest from array of 8 bit numbers.

COMP 4.3 MICROPROCESSORS AND INTERFACING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.3	Microprocessors and Interfacing	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An in-depth understanding of the Intel 8086 architecture and programming model.
2. An ability to write Assembly language programs for a given task.
3. An understanding of different types of memories, peripheral IC's like 8255, 8259 and 8254and their interfacing with the processor.
4. An ability to interface various I/O devices with the processor.

Course Outcomes:

The student after undergoing this course will be able to:

1. Describe the architecture and explain the working of each block in 8086 processor.
2. Analyze the instruction set of 8086 processor.
3. Analyze the timing sequence of various instructions.
4. Create Assembly language programs for a given task.
5. Explain the basic programmable ICs like 8255, 8259 and 8254.
6. Design interfacing of memories and various I/O devices with the processor.

UNIT -1 **(12 Hours)**

The 8086 Microprocessor family overview, 8086 Internal Architecture. Introduction to programming the 8086. 8086 Family Assembly Language Programming: Program Development steps, Constructing the machine codes for 8086 Instructions, Writing programs for use with an Assembler, Assembly Language Program Development Tools

Implementing Standard Program Structures in 8086 Assembly Language: Simple Sequence programs, Jumps, Flags, Conditional Jumps, If –Then, If-Then-Else and Multiple If-Then-Else programs, While-do programs, Repeat-Until Programs. 8086 Instruction Descriptions and Assembler Directives: Instruction Description, Assembler Directives.

UNIT -2 (12 Hours)

Strings, Procedures and Macros: The 8086 String Instructions, Writing and using Procedures, Writing and using Assembler Macros. 8086 System Connections Timing and Troubleshooting: A Basic 8086 Microcomputer System, An example minimum mode System, The SDK-86. 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.

UNIT -3 (12 Hours)

Digital Interfacing: Programmable Parallel Ports and Handshake Input Output, 8255A Programmable Parallel Interface- Internal Block diagram and System Connections, 8255A Operational Modes and Initializations, 8255A Handshake Applications Examples. Cache Memories Coprocessor: 8086 Maximum mode, Direct Memory Access (DMA) Data transfer, Interfacing and Refreshing Dynamic RAMS, A Coprocessor- The 8087 Math Coprocessor.

UNIT -4 (12 Hours)

The 80286,80386 and 80486 Microprocessors: Multiuser/Multitasking Operating System Concepts, The Intel 80286 Microprocessor, The Intel 80386 32 bit Microprocessor. The 80286, 80386 and 40386 Microprocessors: The Intel 80486 Microprocessor. An Introduction to the Pentium Processors.

Recommended Readings:

1. Douglas V. Hall; Microprocessors and Interfacing; TMH,Revised Second Edition.
2. John F. Uffenbeck; The 8086/8088 family design, programming and interfacing; (PHI).
3. Liu and Gibson; Microprocessor Systems: The 8086/8088 family architecture programming and design; PHI.
4. Richard C. Detmer Jones and Bartlett; Introduction to 8086 Assembly Language And Computer Architecture; Publishers ISBN 0-7637-1773-8.
5. Barry B. Brey; The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium and Pentium preprocessor architecture, Programming and Interfacing; PHI.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. 8086 ALP to evaluate the expression.
2. 8086 ALP to find sum and average of n numbers.
3. 8086 ALP to find even and odd number.
4. 8086 ALP to implement linear search.
5. 8086 ALP to find the square of a number using macro and procedure.
6. 8086 ALP to implement bubble sort .
7. 8086 ALP to compare two strings.
    a. Using string instructions.
    b. Without using string instructions.
8. 8087 ALP to evaluate the expression.
9. 8087 ALP to evaluate the identity $\sin^2\emptyset+\cos^2\emptyset=1$.
10. 8087 ALP to evaluate to compute standard deviation.
11. 8087 ALP.
    a.To compute X^y .
    b.To evaluate the expression.

COMP 4.4 DATA STRUCTURES AND ALGORITHMS-II

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.4	Data Structures and Algorithms-II	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

The subject aims to provide the student with:

1. An ability to use data structures as the foundational base for computer solutions to engineering problems.
2. An understanding of the different logical relationships among various data items.
3. Ability to understand the generic principles of computer programming as applied to sophisticated data structures.
4. An ability to plan, design, execute and document sophisticated technical programs to handle various sorts of data structures.

Course Outcomes:

The student after undergoing this course will be able to:

1. Demonstrate the use of complex data structures like trees and graphs.
2. Explain the applications of tree and graph structures in Engineering.
3. Apply the knowledge of data structures to a given problem.
4. Explain the storage management schemes.

UNIT -1 (12 Hours)

Introduction to Trees: Terminology, Types of Binary trees. Array and Linked List representation of binary trees in memory: Traversal in Binary Tree: Non recursive traversal in binary tree: Preorder, in-order, post-order and Level order traversal. Creation of binary trees from the traversal of binary trees. Binary search tree: Traversal in binary search tree, searching in a binary search tree, finding nodes with minimum and maximum key, insertion and deletion in a binary search tree. Threaded Binary Tree: Finding in-order successor and predecessor of a node in threaded tree. Insertion and deletion in threaded binary tree.

UNIT - 2 (12 Hours)

AVL Tree**:** Searching and traversing in AVL trees. Tree Rotations: Right Rotation, Left Rotation. Insertion and Deletion in an AVL Tree. Red Black Trees: Searching,

Insertion and Deletion. Heap: Insertion, Deletion, Building a Heap, Selection algorithm, implementation of Priority Queue. Weighted Path Length. Multi-way Search Tree. B-tree: Searching, Insertion, Deletion from leaf node and non-leaf node. B+ Tree: Searching, Insertion, Deletion. Digital Search Tree.

UNIT - 3 (12 Hours)

Introduction to Graphs: Undirected Graph, Directed Graph, graph terminology, Connectivity in Undirected and Directed Graphs. Spanning tree, spanning forest. Representation of graph: adjacency matrix, adjacency list, Transitive closure of a directed graph and path matrix. Traversals: Breadth First Search, Depth First Search. Connected components in a graph. Network flow problems- a simple maximum flow algorithm.

UNIT - 4 (12 Hours)

Applications of Trees: Huffman Tree, Binary Tree Sort, Heap Sort and Huffman Tree.
Applications of Graphs: Warshall's algorithm. Shortest Path Algorithms: Dijkstra's Algorithm, Bellman Ford Algorithm, Floyd's Algorithm. Minimum Spanning Tree: Prim's Algorithm, Kruskal's Algorithm. Topological Sorting. Storage Management: Sequential Fit Methods: First Fit, Best Fit and Worst Fit methods. Fragmentation, Freeing Memory, Boundary Tag Method. Buddy Systems: Binary Buddy System, Fibonacci Buddy System.
Compaction, Garbage Collection.

Recommended Readings:

1. S. K Srivastava, Deepali Srivastava; Data Structures through C in Depth; BPB Publications; 2011.
2. Yedidya Langsam, Moshej Augenstein, Aaron M. Tenenbaum; Data Structure Using C & C++; Prentice Hall of India; 1996.
3. Ellis Horowitz, Sartaj Sahni; Fundamentals of Data Structures; Galgotia Publications; 2010.
4. Alfred V. Aho, John E. Hopcroft, J. D. Ullman; Data Structures and Algorithms; Addison Wesley; 1983.
5. Jean Paul Tremblay, Paul G. Sorenson; An introduction to data structures with applications; Tata McGrawHill; 1984.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Implementation of operations in a binary search tree.
2. Implementation of insertion, deletion and traversal for fully in-threaded binary search tree.
3. Implementation of AVL tree.
4. Implementation of red black tree.
5. Implementation of insertion and deletion in heap.

6. Implementation of operations in a B tree.
7. Implementation of adjacency matrix creation.
8. Implementation of addition and deletion of edges in a directed graph using adjacency matrix.
9. Implementation of insertion and deletion of vertices and edges in a directed graph using adjacency list.
10. Implementation of traversal of a directed graph through BFS.
11. Implementation of traversal of a directed graph through DFS.
12. Implementation of finding shortest distances using Dijkstra's algorithm.

COMP 4.5 SIGNALS AND SYSTEMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.5	Signals and Systems	3	1	2	3	100	25	--	--	--	**125**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of time-domain representation and analysis of signals and systems.
2. An ability to perform frequency-domain representation and analysis using Fourier tools.
3. An ability to perform frequency-domain representation and analysis using Laplace transform.
4. An understanding of sampling theory.

Course Outcomes:

The student after undergoing this course will be able to:

1. Determine the mathematical representation and classification of signals and systems.
2. Determine the response of an LTI system using convolution and classical methods. Analyze system properties based on impulse response.
3. Determine and analyze the responses of LTI system to periodic signals using Fourier series.
4. Determine and analyze the responses of LTI system to arbitrary time signals using Fourier transform.
5. State sampling theory and its application and convolution and correlation of signal.
6. Determine the properties of continuous time signals and system using Laplace transforms.

UNIT -1 (12 Hours)

Signal, Continuous-time signals, Discrete-time Signals, Graphical, Functional, Tabular and Sequence Representation of Discrete-time Signals. Basic Operations on Signals. Classification of Signals. Signals and Vectors: Vector addition, Scalar multiplication, Dot product (inner product), Norm, Distance, Angle, Projection. Vector Space: Orthogonality in vectors. A System. Classification of Systems. Invertibility and Inverse Systems.

UNIT - 2 (12 Hours)

Time domain analysis of discrete-time systems: Solution of difference equations. natural, forced, total response, and impulse response. Representation of discrete-time signals in terms of impulses. Impulse response and convolution sum. Properties of convolution. Convolution of two sequences. Causality, FIR and IIR systems, stability, BIBO stability criterion, step response, correlation of two sequences. Inverse system and deconvolution. Time domain analysis of continuous-time systems: Solution of differential equations. Natural, forced and total response. Representation of a continuous-time signal. Convolution integral, Properties of convolution, Impulse response of interconnected systems. Causality, Graphical Procedure to Perform Convolution, Stability Step Response, Correlation.

UNIT – 3 (12 Hours)

Fourier Series Analysis of Continuous-Time Periodic signals: Fourier Series Representation of Periodic Signals, Evaluation of Fourier Coefficients, Symmetry Conditions, Cosine Representation, Exponential Fourier series, Existence of Fourier Series, Properties of Continuous-time Fourier Series, Power Representation using the Fourier series, Fourier Spectrum, Gibb's Phenomenon. The Continuous-Time Fourier Transform. Development of Fourier Transform. Fourier Transform of some Standard signals. Properties of Fourier Transform. Fourier Transform of a Periodic signal. Modulation. System Analysis with Fourier Transform.

UNIT - 4 (12 Hours)

Signal and System Analysis using the Laplace Transform: Convergence of the Laplace Transform, s-Plane, The Unilateral Laplace Transform, Properties of unilateral Laplace transform. Inversion of Unilateral Laplace Transform, Inversion of the Bilateral Laplace Transform. Solution of Differential Equations using Laplace Transform. Analysis of Electrical Networks using Laplace Transform., Stability, Block Diagram Representation of Summer, Gain, Feedback, Integrator, cascade connection of blocks and parallel connection of blocks. Signal flow graph, System Realization. State Space Analysis. Sampling: Analog to Digital Conversion, Sampling and Aliasing, Impulse sampling, Sampling Theorem, Anti Aliasing Filter, Pulse Sampling, Flat-Top sampling, Signal Reconstruction, Bandpass Signals. Sampling Bandpass Signals.

Recommended Readings:

1. P. Ramesh Babu, R. Anandanatarajan; Signals and Systems; Scitech Publications, 4th Edition; 2006.
2. Rodger E.Ziemer, William H. Tranter, D. Ronald Fannin; Signals and Systems - Continuous and Discrete; Pearson Education, 4th Edition; 1983.
3. Simon Haykin and Barry Van Veen ; Signals and Systems; John Wiley &Sons (Asia)Pvt. Ltd; 2/e.
4. Oppenheim and Willskay with Hamid Nawab ; Signals and Systems; Prentice Hall of India.
5. Linder, Introduction to Signals and Systems; McGraw Hill.
6. Nagrath, Sharan, Rajan and Kumar; Signals and Systems; McGraw Hill.

COMP 4.6 JAVA PROGRAMMING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 4.6	Java Programming	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of how things work in the web world.
2. An understanding of the client-side implementation of web applications.
3. An ability to understand the generic principles of object oriented programming using "Java".
4. An understanding the use of Graphics programming in "Java".
5. An ability to plan, design, execute and document sophisticated object oriented programs to handle different computing problems using "Java".

Course Outcomes:

The student after undergoing this course will be able to:

1. Design algorithms using principles of object oriented programming
2. Demonstrate the use-cases, pseudocode, and an incremental coding plan for a given Problem specification.
3. Explain the operations of common data structures and algorithms.
4. Design a "Java" program to solve a given problem specification.
5. Illustrate stream I/O, Graphics programming and exception handling.

UNIT – 1 (12 Hours)

Fundamentals of Object Oriented Programming, Java Evolution , Overview of Java Language, Constants, Variables and Data Types, Operators and Expressions, Decision Making and Branching, Decision Making and Looping, Classes, Objects and Methods.

UNIT – 2 (12 Hours)

Arrays, Strings and Vectors, Interfaces: Multiple Inheritance, Packages: Putting Classes together.

UNIT – 3 (12 Hours)

Multithreaded Programming, Managing Errors and Exceptions, Applet Programming.

UNIT – 4 (12 Hours)

Graphics Programming, Managing Input/ Output Files in Java, Java Collections.

Recommended Readings:

1. E. Balagurusamy; Programming with Java A Primer; Tata McGrawHill Companies 5th edition.
2. John P. Flynt ;Java Programming; Thomson 2nd.
3. Ken Arnold ;Java Programming Language; Pearson.
4. Hervert schildt; The complete reference JAVA2; TMH.
5. Cay Horstmann; Big Java; 2nd edition; Wiley India Edition.
6. Sachin Malhotra, Saurabh Chaudhary; Programing in Java; Oxford University Press, 2010.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Programs using constructor and destructor.
2. Creation of classes and use of different types of functions.
3. Count the number of objects created for a class using static member function.
4. Write programs on interfaces.
5. Write programs on packages.
6. Write programs using function overloading.
7. Programs using inheritance.
8. Programs using IO streams.
9. Programs using files.
10. Write a program using exception handling mechanism.
11. Programs using AWT.
12. Programs on swing.
13. Programs using JDBC.