

SHIVAJI UNIVERSITY, KOLHAPUR

**REVISED STRUCTURE AND SYLLABII
OF S.E. COMPUTER SCIENCE AND ENGINEERING
W.E.F. 2008-09.**

Semester - III

Sr. No.	Subject	L	T	P	Theory Marks		TW	POE	Oral	Total Marks
					Written	Online				
1	Applied Mathematics	3	1	-	50	50	25	-	-	125
2	Discrete Mathematical Structures	4	2		50	50	25	-	-	125
3	Data Structures	4	-	-	50	50	-	-	-	100
4	Computer Networks - I	4	-	-	50	50	-	-	-	100
5	Digital Systems & Microprocessors	3	-	2	50	50	25	-	25	150
6	Programming Lab - I	2	-	4	-		50	50	-	100
	Total	20	3	6	250	250	125	50	25	700

Semester – IV

Sr. No.	Subject	L	T	P	Theory Marks		TW	POE	Oral	Total Marks
					Written	Online				
1	Automata Theory	3	1	-	50	50	25	-	-	125
2	Computer Networks - II	4	-	2	50	50	25	-	25	150
3	Computer Organization	4	-	-	50	50	-	-	-	100
4	Advanced Microprocessors	4	-	-	50	50	-	-	-	100
5	Software Engineering	3	-	-	50	50	-	-	-	100
6	Programming Lab - II	2	1	2	-		25	50	-	75
7	Mini Project	-	1	2	-		25	-	25	50
	Total	20	3	6	250	250	100	50	50	700

Note:

1. The term work as prescribed in the syllabus is to be periodically and jointly assessed by a team of teachers from the concerned department.
2. In case of tutorials, students of different batches be assigned problems of different types and be guided for the solution of the problem during tutorial session. Problems thus solved be translated into computer programs wherever applicable and executed by respective batches during practical session.
3. The assignments of tutorials and practicals need to be submitted in the form of soft copy and / or written journal.
4. Breakup of term work marks shall be as follows:
 - a. Mid-semester test – 5 marks.
 - b. End-semester test – 5 marks.
 - c. Tutorial assignments and / or practical performance – 15 marks.

5. The theory exam scheme is as under:
- a. All theory papers of SE (CSE) part 1& part 2 of 100 marks will be divided into two parts.
 - i. **Part-A:** 50 marks theory paper similar to the existing theory paper exam. The nature of the questions will be descriptive, analytical and problem solving.
 - ii. **Part-B:** 50 marks computer based exam with multiple choice questions (MCQs) .
 - b. The marks obtained in the individual heads should be added and considered as marks of the respective theory paper out of 100 marks.
 - c. The questions of part-A and part-B will be based on the entire syllabus of the respective subjects.
 - d. The questions in part-B will be of 1 or 2 marks only.
 - e. Duration of part-A exam will of 2 hours and that of part-B will be of 1 hour.
 - f. The passing scheme for the subject will be similar to existing scheme.
 - g. No separate passing head for part-A and part-B.
 - h. The scheme of revaluation is not applicable for part-B, however is applicable for part-A
 - i. All the existing ordinances will be applicable for passing criteria.

S.E. (Computer Science and Engineering) Semester – III

1. Applied Mathematics - I

Lectures : 3 hrs / week
Tutorials : 1 hr / week

Theory : 100 marks
Termwork : 25 marks

Section - I

1. Statistics: Coefficient of correlation and lines of regression of bivariate data, Fitting of Curves by method of Least-squares. (6)
2. Probability: Random variable, Binomial, Poisson and Normal distributions. (6)
3. Queuing Theory: Introduction, Queuing systems, Distributions in queuing systems, M/M/1 and M/M/S models. (6)

Section – II

4. Introduction to Fuzzy sets, Basic types of fuzzy sets, standard fuzzy set operations, crisp sets, crisp sets versus fuzzy sets. (7)
5. Fuzzy Arithmetic. (5)
6. Constructing Fuzzy sets and operations on fuzzy sets, Fuzzy systems and applications. (6)

Reference Books:

1. A text book of Applied Mathematics: Vol. I, II and III by J. N. Wartikar & P. N. Wartikar , Vidyarthi Griha Prakashan, Pune.
2. Higher Engineering Mathematics by Dr. B. S. Grewal.
3. Operations Research by S. D. Sharma
4. Fuzzy sets and Fuzzy Logic by George J. Klir, Bo Yuan. (Chapters : 1, 2, 3, 4, 10 and 12 upto 12.3 only)
5. Probability and Statistics for Computer science by James L. Johnson (Wiley Student Edition).
6. Probability and Statistics for Engineers by Richard A. Johnson (Pearson Education).
7. Probability, Statistics and Random Processes by T. Veerarajan (TMGH) 2nd Edition.

Term work:

1. For the term work of 25 marks, batch-wise tutorials are to be conducted. The number of students per batch should be as per university pattern for practical batches.
2. Minimum number of assignments should be 10 covering all topics.
3. Assignments on **application to computer science** should be given to the students from the **book mentioned at serial no. 5** in the above list of reference books.

Nature of Question paper:

1. There will be two sections carrying 50 marks each.
2. Each section will have four questions & three questions should be attempted from each.

2. Discrete Mathematical Structures

Lectures : 4 hrs / week

Tutorials : 2 hr / week

Theory : 100 marks

Termwork : 25 marks

Section – I

1. **Mathematical logic:** Introduction, statements and notations, connectives – negation, conjunction, disjunction, conditional, bi-conditional, Statement formulas and truth tables, well formed formulas, Tautologies, Equivalence of formulas, Duality law, Tautological implications, functionally complete sets of connectives, other connectives, Normal and principal normal forms, completely parenthesized infix and polish notations, Theory of inference for statement calculus – validity using truth table, rules of inference, consistency of premises and indirect method of proof. (9)
2. **Set theory:** Basic concepts of set theory, types of operations on sets, ordered pairs, Cartesian product, representation of discrete structures, relation, properties of binary relations, matrix and graph representation, partition and covering of set, equivalence relation, composition, POSET and Hasse diagram, Function – types, composition of functions, Inverse function.(10)
3. **Algebraic systems:** Semigroups and Monoids, properties and examples. (4)

Section – II

4. **Groups:** Definition and examples, subgroups and homomorphism, Group codes – communication model, Generation of codes using checksum, error recovery in group codes. (5)
5. **Lattices and Boolean algebra:** Lattice as POSETs , definition , examples and properties, Lattice as algebraic systems, Special lattices, Boolean algebra definition and examples, Boolean functions, representation and minimization of Boolean functions . (8)
6. **Graph theory:** Basic concepts of graph theory, Storage representation and manipulation of graphs, Fault detection in combinational switching circuits – Faults in combinational circuits, Notions of Fault detection, Algorithm for fault matrix, PERT and related techniques. (9)

Text Book:

1. Discrete mathematical structures with application to computer science
- J. P. Tremblay & R.. Manohar (MGH International)

Note: Scope of the articles mentioned in the syllabus is as per the text book.

References:

1. Discrete mathematics - Seymour Lipschutz, Marc Lipson (MGH), Schaum's outlines.
2. Discrete mathematics and its applications - Kenneth H. Rosen (AT&T Bell Labs)
(mhhe.com/rosen)
3. Schaums solved problem series - Lipschutz
4. Discrete Mathematical Structures – Bernard Kolman, Robert Busby, S.C. Ross and Nadeem-ur-Rehman (Pearson Education).

Term Work: It should consist of minimum 10 to 12 assignments based on following guidelines.

In tutorial session, students of different batches be assigned –

- a) Different exercise problems and be guided for the solution of the problems AND
- b) To write programs in C language on any 4 to 5 following related topics.

1. Generating truth table of a statement
2. Application of bit representation of sets and operations on sets or relations.
3. Conversion of polish expressions.
4. Obtaining the path matrix, paths of different lengths.
5. Different tree traversal methods.
6. Evaluating polynomial expressions using linked lists.
7. Allocation graphs and deadlock detection.
8. PERT related techniques.

3. Data Structures

Lectures : 4 hrs/week

Theory : 100 marks

Section - I

1. **Stacks, Queues, Circular queues:** Definitions, representation, priority queues, operations and their applications. (7)
2. **Searching and Sorting Techniques :**Linear search, Binary search, , Bubble sort, insertion sort, Merge sort, Quick sort, Selection sort, Radix sort, Heap sort. (9)
3. **Lists :** Definition, Representation , Operations and Applications of singly linked list, doubly linked list, circular linked list. (7)

Section – II

4. **Hashing :** Definition, Hash functions, Overflow, Collision, Open Hashing, closed hashing, Rehashing Techniques. (6)
5. **Trees:** Basic Technology, Binary Tree, Traversal methods, Binary search tree, B tree, B+ tree, Heaps - operations and their applications. (10)
6. **Graphs:** Basic concepts of graph theory, storage representation and manipulation of graphs, Introduction to Sparse matrix, representation of sparse matrix using linked list. (7)

Text Books:

1. Data Structure using C -- A. M. Tanenbaum, Y. Langsam, M. J. Augenstein (PHI)
2. Theory and Problems of Data structures -- Lipschutz (MGH International)
3. Data Structure using C -- ISRD Group (TMH) ACE series.

Reference Books :

1. Data structures and Algorithms -- Alfred V. Aho, John E. Hopcroft, J. D. Ullman (Addison- Wesley Series)
2. Data structures -- Seymour Lipschutz (MGH) Schaum's Outlines.
3. Introduction to Data Structures in C – Ashok N. Kamthane (Pearson Education).

4. Computer Networks - I

Lecturers : 4 Hrs/Week

Theory : 100 Marks

Section - I

1. **Introduction** : Data Communications, Networks, The Internet, Protocols and Standards. (4)
2. **Data and Signals** : Analog and Digital, Periodic analog signals, digital signals, Transmission Impairments, Data rate limits and Performance. (6)
3. **Digital Transmission** : Line coding and line coding schemes, Transmission models. (3)
4. **Multiplexing** : Frequency Division Multiplexing, Wavelength Division Multiplexing and Time Division Multiplexing. (3)
5. **Transmission Media** : Guided Media , Unguided Media. (4)
6. **Switching** : Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks. (3)

Section - II

7. **Network Models** : Layered Tasks, The OSI model, Layers in the OSI model, TCP/IP protocol Suite, ATM Reference model. (4)
8. **Data Link Control** : Framing, Flow and error control, protocols - Noiseless channels, Noisy Channels, HDLC, point to point protocols. (8)
9. **Multiple Access** : Random Access, Controlled Access. (5)
10. **Wired LANs : Ethernet** - IEEE Standards, Standard Ethernet, Changes in the standard, Fast Ethernet. (5)

Text Book :

1. Data Communication and Networking -- Behrouz A Forouzan (The McGraw-Hill Companies) 4th Edition.

Reference Books :

1. Data and Computer Communications -- William Stallings (Pearson Education) 7th Edition.
2. Computer Networks -- Andrew S. Tanenbaum (Pearson Education) 4th Edition.

5. Digital Systems and Microprocessors

Lectures : 3 hrs / week
Practical : 2 hr / week

Theory : 100 marks
Term work : 25 marks
Oral : 25 marks

Section – I

1. **Fundamental Concepts:-** Analog and digital systems, Digital and logic circuits, Basic logic operations and gates- OR, AND, NOT. Describing logic circuits algebraically, implementing circuit from Boolean expression. NOR and NAND gates. Boolean theorems, De Morgan's theorems, Universality of NAND & NOR gate. (3)
2. **Flip-flops, Registers and counters:-** Flip-flop using NOR and NAND gates, clocked flip-flops, Clocked S-R, J-K, D Flip-flops, Data storage and transfer, Shift register, Asynchronous counter using Flip-flop. (4)
3. **Binary Arithmetic:-** Binary addition, Signed numbers, Addition and Subtraction in 2's Complement system, overflow, multiplication and division of binary numbers, BCD addition, Hexadecimal addition and subtraction, Full adder. (4)
4. **Microprocessor Architecture and Microcomputer System:-**Microprocessor Architecture and its operation- Microprocessor initiated operations, internal operation, and Peripheral operation. Memory- Flip-flop or latch as storage element, memory map and addresses, memory and instruction fetch, memory classification. Input and output devices,[example of a micro computer system, logic devices used for interfacing- Tri-State devices, buffer, decode, encoder. (7)

Section:-II

5. **8085 Microprocessor Architecture:-** The 8085 MPU, Microprocessor communication and bus timing, De-multiplexing address and Data bus, Generating control signals, The 8085 Architecture, 8085 based microcomputer-machine cycles and bus timing, op-code fetch machine cycle, memory read and write machine cycle. Memory interfacing-memory structure, basic concepts in memory interfacing. (6)
6. **8085 assembly language programming:-**The 8085 programming model, instruction classification, instruction and data format, Writing and execution assembly language program. The 8085 instruction-data transfer operations, addressing modes, Arithmetic operation, Flag concept and cautions, Logic operations, Branch operations. (6)
7. **Stack and interrupt:-** Stack, Instruction related to stack, Important concept in stack, execution of CALL and RET. The 8085 interrupt, RST instructions, vectored interrupts, RIM and SIM instructions. (4)
8. **Interfacing I/O devices:** Basic interfacing concepts, peripherals i/o instructions - IN, OUT, I/O execution, device selection and data transfer, Input interfacing, Interfacing i/o using decoder, interfacing displays, memory mapped i/o. (2)

Text Books:

1. Digital systems, principles and applications – Ronald Tocci, Neal Widmer, Gregory Moss (Pearson Education) 9th Edition.
2. Microprocessor Architecture – programming and applications with 8085 – Ramesh Gaonkar (Penram International) 4th Edition.

Term Work: It should consist of minimum 8 to 10 experiments on the above topics.

6. Programming Laboratory – I

Lectures : 2 hrs / week

Practical : 4 hrs / week

Term work : 50 marks

POE : 50 marks

1. An Overview of C : Compilers vs. Interpreters, The Form of a C Program, The Library and Linking, Separate Compilation, Compiling a C Program, C's Memory Map; Expressions - The Basic Data Types, Modifying the Basic Types, Identifies Names, Variables, The Four C Scopes, Type Qualifiers-const, volatile, Storage Class Specifiers; Statements - Selection Statements, Iteration Statements, Jump Statements, Expression Statements, Block Statements; Arrays and Strings- Two-Dimensional Arrays, Arrays of Strings, Multidimensional Arrays, Array Initialization, Variable-Length Arrays. (5)

2. Pointers: What Are Pointers?, Pointer Variables, The Pointer Operators, Pointer Expressions, Pointers and Arrays, Arrays of Pointers, Multiple Indirection, Initializing Pointers, Pointers to Functions and structures, C's Dynamic Allocation Functions, restrict-Qualified Pointers, Problems with Pointers (4)

3. Functions : The General Form of a Function, Understanding the Scope of a Function, Parameter passing, Passing arrays to functions, Function Arguments, argc and argv-Arguments to main(), The return Statement, What Does main() Return?, Recursion, Function Prototypes, Declaring Variable Length Parameter Lists, The "Implicit int" Rule, Old-Style vs. Modern Function Parameter Declarations, The inline Keyword. (4)

4. Structures, Unions, Enumerations, and typedef : Structures, Arrays of Structures, Passing Structures to Functions, Structure Pointers, Arrays and Structures Within Structures, Unions, Bit-Fields, Enumerations, Using sizeof to Ensure Portability, typedef. (4)

5. Console I/O : Reading and Writing Characters, Reading and Writing Strings, Formatted Console I/O, Printf(), scanf(), Suppressing Input. (2)

6. File I/O: File I/O, Standard C vs. Unix File I/O, Streams and Files, File System Basics, fread() and fwrite(), fseek() and Random-Access I/O, fprintf() and fscanf(), The Standard Streams. (3)

7. The Preprocessor and Comments : The Preprocessor, #define, #error, #include, Conditional Compilation Directives, #Undef, Using defined, #line. (2)

Text Book:

1. C The Complete Reference – Herbert Schildt (Tata McGraw-Hill Edition)

References:

1. C The programming language – Kernigham and Ritchie.
2. Understanding pointers in C - Yashavant Kanetkar (BPB)
3. Programming with C – Byron Gottfried (TMGH) Schaum series.
4. Programming in ANSI C – E. Balagurusamy (TMGH).

Term Work:

It should consist of minimum 14 experiments based on the following guidelines and should be conducted in Unix/Linux platform.

1. Implement matrix operation by representing matrix in the form of (a) array (b) linked list. Matrix operation like – Multiplication of matrices, finding the Inverse matrix, singular matrix, upper triangle, lower triangle, symmetric matrix, skew symmetric, triangular matrix, etc.
2. Implement a C program that will accept a hexadecimal number as input and then display a menu that will permit any of the following operation to be carried out.
 - (a) Display hex equivalent of one's complement.
 - (b) Carry out a masking operation and then display the hex equivalent of the result.
 - (c) Carry out a bit shifting operation and then display the hex equivalent of the result.
 - (d) Exit.

If the masking operation is selected, prompt the user for the type of operation (bit wise and bit wise exclusive or bit wise or) and then a (hex) value for the mask. If the shifting operation is selected, prompt the type of shift (left or right) and then the no. of bits to be shifted. Test the program with several different (hex) input values of your own choice.

Note: Conversion of different equivalent forms like – binary, octal, decimal and others can also be performed and tested.

3. Define a mask and write the appropriate masking operating for each of the situations described below:
 - a. Copy the odd bits (bits 1, 3, 5.....15) and place zero in the even-bit location (bit 0, 2, 4, 14) of 16 bit, unsigned integer quantity represented by the variable v. Assume that bit 0 is the rightmost bit.
 - b. Strip the msb (the leftmost bit) from an 8-bit character represented by variable c (certain word processor use this bit to control the formatting of the text within a document. Stripping this bit i. e. setting it to zero, can transform the word processor documents into a text file consisting of ordinary ASCII characters.)

- c. Copy the odd bits (bits 1, 3, 5, ..., 15) and place one's in the even bit locations (bits 0,2,4, ...,14) of a 16 bit unsigned integer quantity represented by variable v. Assume bit 0 is the rightmost bit.
 - d. Toggle (invert) the values of bits 1 & 6 of the 16 bit unsigned integer quantity represented by variable v, while preserving all the remaining bits. Assign new bit pattern to v.
4. Implement and compare liner and binary search for any given huge data set <min – 10000>. Data must be float, string.
5. Implement any 2-3 sorting techniques and find the number of comparison required to sort random data –set of around 10000.
6. Write a problem which
 - a. Implements pointers to function
 - b. has a function having parameters as pointer to function.
7. Implement stack using array and linked list.
8. Implement queue, priority queue, circular queue using array and linked list.
9. Write a program which shows advantages of
 - (a) Static variable
 - (b) Static function (using multiple C files)
 - (c) volatile
 - (d) Extern (using multiply C files.)
10. Implement hashing and rehashing operating on data like float and string.
11. Implement doubly linked list, Circular linked list, doubly circular linked list.
12. Implement and perform different operation of binary tree, B-tree insertion, deletion, modification, finding the depth of the tree.
13. Create your own library file and header file.
14. Implement all loops. Also implement equivalent loops –for, while, do-while using recursion.
15. Implement Towers of Hanoi and Ackermann's function.

S.E. (Computer Science and Engineering) Semester – IV

1. Automata Theory

Lectures : 3 hrs / week

Tutorial : 1 hr / week

Theory : 100 marks

Term work : 25 marks

Section – I

1. Recursive Definitions ,Definition and types of grammars and languages, Regular expressions and corresponding regular languages , examples and applications, unions, intersection & complements of regular languages , Finite automata definition and representation , Non-deterministic F.A. , NFA with null transitions, Equivalence of FA's , NFA's and NFA's with null transitions . (6)
2. Kleene's theorem – Part I & II statements and proofs, minimum state FA for a regular language , minimizing number of states in FA . (3)
3. Grammars and languages – Derivation and ambiguity , BNF and CNF notations , Union, Concatenation and * 's of CFLs , Eliminating production and unit productions from CFG , Eliminating useless variables from a Context Free Grammar . (6)
4. Parsing – Top-Down , Recursive Descent and Bottom-Up Parsing. (3)

Section – II

5. Push Down Automata – Definition , deterministic PDA & types of acceptance, equivalence of CFGs & PDAs. (4)
6. CFL's and non CFL's – Pumping Lemma and examples , intersection and complements. (3)
7. Turing Machines-models of computation, definition of TM as Language acceptors, combining Turing machines, computing a function with a TM. (5)
8. Variations in TM- TMs with doubly-infinite tapes, more than one tape, Non-deterministic TM and Universal TM. (6)

Text Books :

1. Introduction to languages & theory of computations – John C. Martin (MGH)
2. Discrete Mathematical Structures with applications to computer science – J.P. Trembley & R. Manohar (MGH)

References:

1. Introduction to Automata Theory , Languages and computation – John E. Hopcraft , Rajeev Motwani , Jeffrey D. Ullman (Pearson Edition) .
2. Introduction to theory of computations – Michael Sipser (Thomson Brooks / Cole)
3. Finite Automata and Formal Languages – Linz
4. Introduction to Theory of Computer Science, Languages and machines (III Edition) - Sundkumar

Term work: It should consist of minimum 10 to 12 assignments based on the topics of the syllabus and exercise problems mentioned in the text books.

2. Computer Networks - II

Lectures : 4 hrs / week

Practical : 2 hrs / week

Theory : 100 marks

Term work : 25 marks

Oral : 25 marks

Section – I

1. **The Network Layer :** Network Layer Design Issues, Routing Algorithms – Optimality Principle, Shortest path Routing, Flooding, Distance Vector Routing, Link State Routing. (6)
2. **Logical addressing :** IPv4 Addresses , IPv6 Addresses. (6)
3. **Congestion Control :** General Principle, Prevention Policies, Congestion control in virtual circuit subnet, congestion control in datagram subnet, Load Shedding, Jitter control. (6)
4. **Quality of Services :** Requirements, Techniques for achieving good quality service, Integrated services. (4)

Section – II

5. **Quality of Services :** Differentiated services . (2)
6. **The Transport Layer :** The Transport Service, Elements of Transport Protocols, Internet Transport Protocols - UDP, TCP. (10)
7. **Security :** Cryptography - Traditional Ciphers, RSA. (5)
8. **Application Layer :** Name Space, Domain Name Space, Distribution of Name Space, DNS in the Internet, Resoution, DNS messages, Remote Login, Electronic Mail, File Transfer, WWW and HTTP. (5)

Text Books :

1. Computer Networks -- Andrew S. Tanenbaum (Pearson Education) 4th Edition (Refer chapter – 1,3,4,5 and 6)
2. Data Communication and Networking -- Behrouz A Forouzan (The McGraw-Hill Companies) 4th Edition (Refer chapter – 2,6 and 7).

Reference Books :

1. Computer Networks - Principles, Technologies and Protocols for Network Design Natalia Olifer and Victor Olifer (Wiley India Edition).

Term Work: It should consist of minimum 10 programming assignments based on the following.

1. File transfer using RS-232
2. File transfer using Stop and Wait Protocol / Go back n / Selective Repeat Protocol
3. Implementation of Hamming code / CRC for error detection / recovery.
4. Implementation of Shortest Path algorithm
5. Study of TCP/IP network
6. File transfer using TCP
7. File Transfer using UDP
8. Data transfer application using TCP/IP protocol suite.
9. Use of DNS client utilities – Nslookup and Dig
10. Implementation of cryptographic algorithms.

3. Computer Organization

Lectures :4 hrs / week

Theory : 100 marks

Section – I

1. **Basic Computer Organization :** Evolution of computers - Mechanical era, Electronic computers, Generations, VLSI era, CPU organization , communications, user and supervisor modes, accumulator based CPU, System bus, instruction cycle, types of instruction(zero, one, two and three address machines), IO interface, RISC & CISC, definition, comparison and examples. (6)
2. **CPU design:** Specifications, (memory, speed, frequency etc.) with example, Instruction fetching, decoding, executing, Case Study (architecture, block diagram, instruction sets etc.), Pentium 4 processor, AMD processor. (4)
3. **Computer Arithmetic:** Data Representation, basic formats, storage order, fixed point numbers, binary, signed, decimal, hexadecimal, Floating point numbers, basic formats, normalization, biasing, IEEE754 format, Fixed point arithmetic - Addition and subtraction, overflow, high speed adders, adder expansion, Fixed point multiplication - Two's complement multiplier, Booth's algorithm, Combinational array multiplier, Fixed point division - Restoring, Non restoring algorithm, Combinational array divider, Division by repeated multiplication, Floating point arithmetic - Basic operations, Difficulties, Floating point units, Addition, subtraction, multiplication, division. (12)

Section - II

4. **Control Design:** Introduction, multi cycle operation, implementation methods, Hardwired control, design methods, state tables, GCD processor, Classical method, one hot method, Design example- twos complement multiplier control, CPU control unit design. (6)
5. **Micro programmed control:** Basic concepts, control unit organization, parallelism in microinstructions, Microinstruction addressing, timing, Control unit organization, Design example- twos complement, multiplier control, Control field encoding, encoding by function, multiple microinstruction formats. (6)

6. **Memory Organization:** Types of memory, Memory systems, multilevel, address translation, memory allocation, Caches, Associative memory, direct mapping, set associative addressing. (10)

Text Books :

1. Computer Architecture and Organization - John P Hayes (MGH) 3rd Edition.
2. Computer Systems Organization & Architecture - John D. Carpinelli (Pearson Education)

References:

1. Computer Organization - Hamacher Zaky (MGH).
2. <http://cse.stanford.edu/class/sophomore-college/projects-00/risc/riscisc/> (RISC vs CISC)
3. <http://www.cpu-world.com/sspec/>
4. http://www.intel.com/technology/itj/q12001/pdf/art_2.pdf (The Micro architecture of the Pentium 4 Processor)
5. http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/30579_AMD_Processor_Evaluation_Guide3.1.pdf (AMD Processor Performance Evaluation Guide).

4. Advanced Microprocessors

Lectures : 4 hrs / week

Theory : 100 marks

Section – I

1. **The Processors: 8086/8088 –Architectures, pin Diagrams and Timing Diagrams:** Register organization of 8086, Architecture, Signal descriptions of 8086, Physical memory organization, General bus operation, I/O addressing capability, Special Processor activities, Minimum mode 8086 System and timings, Maximum mode 8086 System and timings, The Processor 8088. (5)
2. **8086/8088 Instruction Set and Assembler Directives:** Machine language Instruction Formats, Addressing modes 8086, Instruction set of 8086/8088, assembler directives and operators. (2)
3. **The Art of Assembly Language Programming With 8086/8088:** A few machine level programs, Machine coding the programs, Programming with an Assembler, Assembly language example programs (3)
4. **Special Architectural Features and Related Programming:** Introduction to stack, Stack structure of 8086/88, Interrupts and interrupt service routines, Interrupts cycle of 8086/8088, Non mask-able interrupt, Mask-able interrupt(INTR). (5)
5. **80286-80287 -- A Microprocessor With Memory Management and Protection:** Salient features of 80286, Internal Architecture of 80286, Signal description of 80286, Real addressing modes, Protected virtual address mode (PVAM), Privilege, Protection. (5)

Section:-II

6. **80386—80387 and 80486 the 32-Bit Processors:** Salient features of 80386DX, Architecture and Signal descriptions of 80386, Register organization of 80386, Data types of 80386, Real address mode of 80386, Protected mode of 80386, Segmentation, Paging, Virtual 8086 mode. (10)
7. **Recent Advances in microprocessor Architectures—A Journey from Pentium Onwards:** Salient features of 80586(Pentium), A few relevant concepts of computer architecture, system architecture, Branch prediction, Enhanced instruction set of Pentium, What is MMX?, Intel MMX Architecture, MMX data type, Salient points about multimedia application programming, Journey of Pentium-Pro and Pentium- II, Pentium-III (P-III)---The CPU of the next millennium. (9)
8. **Pentium-4 -- processor of the New Millennium:** Genesis of birth of Pentium-4, Salient features of Pentium-4, Netburst micro architecture for Pentium-4, Instruction Translation Look aside Buffer (ITLB) and branch prediction, Why out of order execution, Rapid execution module, Memory Subsystem, Hyper-threading technology, Hyper-threading in Pentium . (5)

Text book:-

1. Advanced Microprocessors And Peripherals -- A.K.Roy, K.M.Bhurchandi (TMGH) 2nd Edition.

Reference books:-

1. Microprocessors and Interfacing -- Douglas V Hall (TMGH) Revised 2nd Edition.
2. Microcomputer system – The 8086/8088 family –Liu & Gibson (PHI)
3. Advanced 80836 programming –Turley (TMH)
4. Intel 8086, 80836 Manual , www.intel.com.
5. Assembly Language for Intel-Based Computer - Kip R. Irvine (Pearson Education) 4th Edition.

5. Software Engineering

Lectures : 3 hrs / week

Theory : 100 marks

Section – I

1. **Introduction:** The S/W problem, S/W Engg. Problem, The S/W Engg. approach. (2)
2. **Software Processes:** Software Process, Characteristics of a software process, Software development process, project management process, Software configuration management process, process management process. (4)
3. **Software requirements analysis and specification:** S/W requirements, problem analysis, Requirements Specification, validation, metrics. (3)
4. **Planning a Software Project:**
 - a. Cost estimation, project scheduling, staffing and personnel planning,

- b. Software Configuration Management plans, Quality Assurance plans.
 - c. Project Monitoring Plans, Risk Management. (5)
5. **Function oriented design:** Design principles, module level concepts, Design notation and specification Structured Design methodology, Verification, Metrics. (4)

Section – II

6. **Object Oriented Design :** OO Analysis & Design, OO Concepts, Design concepts, UML, A Design Methodology, Metrics. (4)
7. **Coding:** Programming Practice, verification, Metrics. (3)
8. Testing: Testing Fundamentals, Black box, White box testing, testing process, Metrics for estimation, A Reliability Model. (5)
9. **Software Quality:** objectives, need for improvement, cost of Quality, Software quality factors, Total Quality Management, Quality standards such as ISO, CMM and CMMI along with their comparison. (3)
10. The project management plan: Team management, customer communication and issue resolution, the structure of the project management plan. (3)

Text Book:

1. An integrated approach to S/W Engineering - Pankaj Jalote. (Narosa Publishers) 3rd Edition.

References:

1. Software Project Management in practice – Pankaj Jalote.(Pearson Education)
2. Software Engineering. : Practitioner’s Approach – Roger S. Pressman (TMGH)
3. Software Engineering - Jawadekar W.S. (TMGH)
4. Object oriented software concepts - Bertrand Mayer.

6. Programming Laboratory - II

Lectures : 2 hrs / week

Tutorial : 1 hr /week

Practical : 2 hrs / week

Term work : 25 marks

POE : 50 marks

1. **An Overview of C++:** The Origins of C++, Encapsulation, Polymorphism, Inheritance, Function Overloading, Operator Overloading, Constructors & Destructors, C++ key words. (2)
2. **Classes & Objects:** Relation of Classes, Structures & Union, Friend Functions, Friend Classes, Inline Functions, Parameterized constructors, Static class members, Scope resolution operators, Passing objects to functions, nested classes, and local classes. (3)

3. **Arrays, Pointers, Dynamic Allocation Operators:** Arrays of objects, Pointers to objects, Type checking C++ Pointers, This Pointer, Pointers to derived types, Pointers to class members, Dynamic allocation operators- new & delete operators. (3)
4. **Function Overloading, Operator Overloading, Copy Constructors & Default Arguments:** Function overloading, Overloading constructor function, copy constructors, Operator overloading using friend function, Overloading new & delete operators, overloading some special operators like [],(),->,Comma operator. (3)
5. **Inheritance:** Single Inheritance, multilevel Inheritance, multiple Inheritance, hybrid Inheritance, hierarchical Inheritance, Virtual base classes. (2)
6. **Virtual Functions & Polymorphism:** Pure virtual function, calling virtual function through a base class, Abstract classes, Early vs. Late binding. (2)
7. **File and Streams:** Streams, String I/O, Character I/O, Object I/O, I/O with multiple objects, File pointers and redirections. (2)
8. **Templates:** Generic classes, Generic functions, Applying generic functions, type name & export keyword, power of templates. (3)
9. **Exception Handling:** Fundamentals, Handling derived class exceptions, exception handling options: catching, throwing & handling of the exception. (2)
10. **Overview of Stream classes, Run-Time Type ID, Namespaces & Standard Template Library:** C++ streams, C++ stream classes, RTTI, Namespace fundamentals, STL containers, STL algorithms, STL iterators. (2)

Text Book:

1. The Complete Reference: C++ - Herbert Schildt (Tata McGraw-Hill Edition)

Reference Books:

1. Object oriented programming in Turbo C++ - Robert Lafore(Galgotia)
2. Programming with C++ - D. Ravichandran(TMGH)

Term work:

It should comprise of minimum 10 experiments. Students of different batches should implement different programs based on the following guidelines in **UNIX/ Linux** platform.

(A) 4-6 assignments should consist of implementing ALL following concepts-

Constructor, Destructor, Function overloading, Constructor overloading, Operator overloading, Multiple inheritance, Multilevel inheritance, Static variables, Function in class, Virtual function, Virtual class, Virtual destructor, Function template, Friend class and function, File handling, Templates, STL

(B) 3-4 assignments on implementing object oriented programs for the problems of Discrete Mathematical Structure of SE-I(CSE), like –

1. Representing a Set in bit form and implementing the set operation like-Union, Intersection, Relative Complement, symmetric difference etc.
2. Conversion of Polish expressions.
3. Obtaining path matrix and paths of different lengths.
4. Evaluating polynomial expression (PE) using linked list and performing operations on PE like Multiplication, addition subtraction, etc.
5. Check dead lock for any given resource allocation graph.

(C) 3-4 assignments on implementing the data structures like-

1. Implement sorting /searching algorithms using function template and virtual function.
2. Implement stack / queue using class template.
3. Implement B/B ++ tree and performing operation on the tree using object oriented concepts
4. Create a linked list as an object. Perform merging of two objects (linked lists) and splitting of object. (Use operator overloading).
5. Implement hashing and rehashing (considering occurrence of overflow).

7. Mini Project - I

Tutorial: 1 hr/ week

Practical : 2 hrs/ week

Term work : 25 marks

Oral : 25 marks

The mini project should be undertaken preferably by a group of 3-5 students who will jointly work and implement the project. The group will select a project with the approval of the guide and submit the name of the project with a synopsis, of the proposed work, of not more than 02 to 03 pages. The mini project could be based upon the problem statements as that of programming contests (**ACM-ICPC** or others) **OR** small application like -

1. Library Management System
2. Inventory Management System
3. Telephone Directory Management

4. Student Database Management
5. Stores Management
6. Student Attendance System
7. Examination Result Analysis System
8. Super Market Billing System

The group is expected to complete analysis of problem, system design as a part of the term work submission in the form of a report. The project must be implemented in C/C++. The term work assessment will be done jointly by a panel of teachers of the department.

The external oral examination will be conducted by the examiners appointed by the University.

BOS in Computer Science & Engineering

Equivalent subjects at SE (CSE) – I & II Pre-revised course to the Revised course of S. E. (CSE) Sem – III & IV.

SE (CSE) Part – I

Sr.no.	SE (CSE) –I (Pre-Revised)	Equivalent / Replacement subject
1	Computational Mathematics	Mathematics for Computer Science
2	Discrete Mathematical Structures	Discrete Mathematical Structures of S.E(CSE) Sem-III (Revised)
3	Data Structures	Data Structures of S.E(CSE) Sem-III (Revised)
4	Switching Theory and Logic Design	Switching Theory
5	Data communications	Computer Networks-I of S.E.(CSE) Sem-III (Revised)
6	Programming Lab-I	Programming Lab-I of S.E(CSE) Sem-III (Revised)

S.E. (CSE) part - II

Sr.no.	SE (CSE) –I (Pre-Revised)	Equivalent / Replacement subject
1	Statistics and Fuzzy Systems	Applied mathematics of S.E (CSE) Sem-III (Revised)
2	Formal Systems and Automata	Automata theory of S.E(CSE) Sem – IV (Revised)
3	Microprocessor - I	Digital Systems and Microprocessors of SE (CSE) Sem – III (Revised)
4	Computer Networks	Computer Networks-II of S.E(CSE) Sem-IV (Revised)
5	Computer Organization	Computer Organization of S.E.(CSE) Sem-IV (Revised)
6	Programming Lab-II	Programming Lab-II of S.E(CSE) Sem-IV (Revised)

The syllabus for the replacement subjects is as given below:

1. Mathematics for Computer Science

Section – I

1. Ordinary differential of first order and first degree; methods of solution: variable separable, homogeneous, non-homogeneous, exact, non-exact, linear, non-linear and Bernoulli's differential equation.
2. Linear differential equations: Linear differential equations with constant coefficients (without method of variation of parameter), homogeneous linear differential equations.
3. Partial differential equations: Four standard forms of Partial differential equations of first order.

Section – II

4. Fourier series: Definition, Euler's formulae, expansion of functions, change of interval, even and odd functions, half range Fourier series.
5. Operation research: Introduction, formulation of LP problems, simplex methods for solutions of LPP.
6. Assignment problem: Introduction, mathematical formulation of assignment problem, method for solving assignment problem.

Reference Books:

1. A text book of applied mathematics, Vol –I and II by J.N. & P.N. Wartikar (Vidhyarthi Griha Prakashan, Pune).
2. Higher Engg mathematics by B.S. Grewal (Khanna Publication, Delhi)
3. Operation research by S.D. Sharma.

2. Switching Theory

Section – I

1. Fundamental concepts: Introduction, digital signals, basic digital circuits, NAND & NOR operations, EX-OR operation, Boolean algebra, practical examples of IC gates, number systems and their conversions – binary, octal, hex, signed numbers, basic characteristics of TTL and CMOS logic families.
2. Flip Flops and Semi-conductor memories: SR flip-flop, JK, D flip flop, excitation table of JK flip flop, clocked flip flop types, basic register, introduction to memory, memory organization and operation, expanding memory size, classification of memories – ROM, RAM, CAM, PLA.
3. Arithmetic circuits: Binary addition, subtraction, 1's and 2's complement arithmetic, carry loop ahead adder, addition & subtraction using 1's and 2's complement, arithmetic logic unit – basic concept, half adder, full adder, applications of arithmetic circuits in fault detection.

Section – II

4. Counters: Register, shift register, counters – asynchronous, up-down, mod counters, 74 series asynchronous counter ICs, design examples, Synchronous counter introduction.
5. Codes: BCD, Gray, Excess 3 code counters, Binary to BCD, Binary to Gray, Gray to binary, 7-segment display device types, Decoder , driver for 7 segment display using 7447.

Reference Books:

1. Modern Digital Electronics – R.P. Jain (TMH)
2. Digital Logic Design – B. Holdsworth (East-West Press).