1. **Network Administrator**
2. **System Administrator**
3. **Director Infrastructure**
4. **Network/Cyber Security Administrator**
5. **Principal Computer Engineer**

Sr. No.	Core Areas	Percentage
1.	Computer Architecture and Organization	1 10%
2.	Electronic Devices and Circuits	5%
3.	Operating Systems	10%
4.	Discrete Structures	5%
5.	Data Structures and Algorithms	5%
6.	Computer Networks and Communications	10%
7.	Programming Paradigms and Language	5%
8.	Software Engineering	5%
9.	Database Management Systems	5%
10.	Digital Logic Design	5%
11.	Microprocessor Systems and Interfacing	5%
12.	Digital System Design	5%
13.	Distributed Systems	5%
14.	Signals and Systems	5%
15.	Embedded Systems	5%
16.	Human Computer Interaction	5%
17.	Social and Professional Issues	5%
	Total	**100%**

Computer Engineering (Detailed)		
Sr. No.	Core Areas	Percentage
1.	**COMPUTER ARCHITECTURE AND ORGANIZATION:** 1.1 Fundamentals of Computer Architecture. 1.2 Computer Arithmetic. 1.3 Memory system organization & architecture. 1.4 Interfacing & Communication. 1.5 Device subsystems. 1.6 Processor systems design. 1.7 Organization of C.P.U. 1.8 Performance & Improvement. 1.9 Multiprocessor System. 1.10 Parallel Programming Models.	10%
2.	**ELECTRONIC DEVICES AND CIRCUITS:** 2.1 Small signal analysis of transistor circuits. 2.2 Analysis of the basic operational amplifier. 2.3 Feedback amplifiers. 2.4 Classification of amplifiers. 2.5 Wave generation circuits. 2.6 Power supply circuits and regulation. 2.7 Diode circuit analysis and application. 2.8 Modeling of amplifiers. 2.9 Operational amplifier application and analysis.	5%

	2.10 Circuit modeling and simulation. **2.11 Data conversion circuits.** **2.12 Elementary Transient Analysis.** **2.13 Sinusoidal State Analysis.** **2.14 Exponential Excitation and the Transformed Network.** 2.14.1 Mutual inductance. 2.14.2 Integrated circuit building blocks. 2.14.3 Circuit modeling and simulation.	
7.	**PROGRAMMING PARADIGMS AND LANGUAGES:** **7.1 Programming Fundamentals.** **7.2 Procedural Programming.** **7.3 Functional Programming.** **7.4 Object oriented design.** **7.5 Encapsulation and information hiding.** **7.6 Event driven programming.** **7.7 Programming constructs.** **7.8 Recursion.** **7.9 Object Oriented Programming (Polymorphism, class hierarchies, collection, classes and iteration protocols, fundamental design patterns).**	5%

8.	**SOFTWARE ENGINEERING:** 8.1 Software requirements and specifications. 8.2 Software design. 8.3 Software testing and validation. 8.4 Software evolution. 8.5 Software tools and environments. 8.6 Software project management. 8.7 Software fault tolerance.	5%
3.	**OPERATING SYSTEMS:** 3.1 History & Overview. 3.2 Design Principles. 3.3 Concurrency. 3.4 Scheduling and dispatch. 3.5 Memory Management. 3.6 Device Management. 3.7 I/O Management. 3.8 Multithreading. 3.9 Distributed Operating Sys.	10%
4.	**DISCRETE STRUCTURES:** 4.1 Functions, relations and sets. 4.2 Basic logic. 4.3 Proof techniques.	5%

	4.4 Basics of counting. **4.5 Graphs and trees.** **4.6 Recursion.**	

9.	**DATA STRUCTURES AND ALGORITHMS:** 9.1 Primitive types. 9.2 Arrays. 9.3 Records. 9.4 Strings and string processing. 9.5 Data representation in memory. 9.6 Static, stacks, and heap allocation. 9.7 Runtime storage management. 9.8 Pointers and references. 9.9 Linked structures. 9.10 Implementation strategies for stacks, queues, and hash tables. 9.11 Implementation strategies for graphs and trees. 9.12 Strategies for choosing the right data structure. 9.13 Concepts of graphs and trees. 9.14 Basic strategies that underpin the design of algorithms. 9.15 Fundamental algorithms for counting, searching, sorting. 9.16 Manipulation of hash tables, symbol tables, queues, trees, and graphs. 9.17 Distributed algorithms. 9.18 Fundamentals of computability theory; relevance to security; relevance of design and analysis of algorithms to software design and implementation.	5%

10.	**COMPUTER NETWORKS AND COMMUNICATIONS:** 10.1 Computer network architecture. 10.2 Computer network protocol. 10.3 Local and wide area networks. 10.4 Client- server computing. 10.5 Data security and integrity. 10.6 Wireless and mobile computing. 10.7 Performance evaluation. 10.8 Network management. 10.9 Compression & decompression. 10.10 Encoding ad modulating (A/D and D/A conversion). 10.11 Interfaces and modems. 10.12 Transmission media. 10.13 Multiplexing. 10.14 Error detection and correction.	10%
11.	**DATABASE MANAGEMENT SYSTEM:** 11.1 Data modeling. 11.2 Relational databases. 11.3 Database query languages. 11.4 Relational database design. 11.5 Transaction processing. 11.6 Distributed databases.	5%

12.	**DIGITAL LOGIC DESIGN:** 12.1 Switching theory. 12.2 Combinational logic circuits. 12.3 Modular design of combinational circuits. 12.4 Memory elements. 12.5 Sequential logic circuits. 12.6 Digital systems design. 12.7 Modeling and simulation. 12.8 Formal verification. 12.9 Fault models and testing.	5%
13.	**MICROPROCESSOR SYSTEMS AND INTERFACING:** 13.1 Overview of microprocessors and microcontrollers. 13.2 Microprocessor system architecture. 13.3 Assembly language fundamentals. 13.4 Interfacing.	5%
14.	**DIGITAL SYSTEM DESIGN:** 14.1 Digital Design Methodology. 14.2 Architectures for basic building blocks. 14.3 Timing and control concepts. 14.4 HW implementation for specific applications. 14.5 Design modeling with a hardware description language. 14.6 Functional and timing simulation of digital systems. 14.7 Implementation in programmable logic devices and	5%

	field-programmable gate arrays. 14.8 Formal verification. 14.9 Fault models. 14.10 Testing.	
15.	DISTRIBUTED SYSTEMS: 15.1 Classification of models: parallel machine models (SIMD, MIMD, SISD, MISD): Flynn's taxonomy, Handler's classification, message passing. 15.2 Granularity, levels of parallelism. 15.3 Multiprocessors and multi-computers: Topology tightly coupled and loosely coupled architectures. 15.4 Processes: threads, clients, servers, code migration, software agents. 15.5 Physical and logical clocks: clock synchronizing algorithms, Lamport timestamps, vector timestamps. 15.6 Election Algorithms. 15.7 Mutual Exclusion algorithms. 15.8 Distributed transactions: models, classification, 15.9 concurrency control.	5%
16.	SIGNALS AND SYSTEMS: 16.1 Time-domain and frequency-domain methods for modeling and analyzing continuous and discrete-data signals and systems. 16.2 Laplace transforms. 16.3 Fourier series and transforms. 16.4 Sampling. 16.5 Discrete signals.	5%

	16.6 Z-transforms. 16.7 Digital filters. 16.8 Digital modulation.	
17.	EMBEDDED SYSTEMS: 17.1 Embedded system overview and fundamentals. 17.2 Embedded design life cycle. 17.3 Sensors and actuators. 17.4 Real time operating systems. 17.5 Low-power computing. 17.6 Reliable system design. 17.7 Design methodologies. 17.8 Embedded multiprocessors. 17.9 Networked embedded systems.	5%
18.	HUMAN COMPUTER INTERACTION: 18.1 Foundations of human-computer interaction. 18.2 Graphical user interface. 18.3 I/O technologies. 18.4 Intelligent systems. 18.5 Human-centered software evaluation. 18.6 Human-centered software development. 18.7 Interactive graphical user-interface design. 18.8 Graphical user-interface programming. 18.9 Graphics and visualization. 18.10 Multimedia systems.	5%

19.	**SOCIAL AND PROFESSIONAL ISSUES:** **19.1 Professional and ethical responsibilities.** **19.2 Risks and liabilities.** **19.3 Intellectual property.** **19.4 Privacy and civil liberties.** **19.5 Computer crime.** **19.6 Economic issues in computing.** **19.7 Philosophical frameworks.**	**5%**
	Total	**100%**

Quality Assurance Expert

Sr. No.	Core Areas	Percentage
1.	Programming Fundamentals	5%
2.	Object Oriented Paradigm	5%
3.	Discrete Structures	5%
4.	Data Structures and Algorithms	5%
5.	Digital Logic and Computer Organization	5%
6.	Operating Systems	5%
7.	Database Systems	5%
8.	Software Engineering & Development	5%
9.	Computer Communication and Networks	5%
10.	Computer Architecture & Assembly Language	5%
11.	Theory of Automata and Formal Languages	5%
12.	Analysis of Algorithms	5%
13.	Artificial Intelligence	5%
14.	System Programming	5%
15.	Numerical Computing	5%
16.	Quality Assurance & Testing	25%
	Total	**100%**

Quality Assurance Expert (Detailed)

Sr. No.	Core Areas	Percentage
1.	**PROGRAMMING FUNDAMENTALS:** Overview of computers and programming. Overview of language. Basics of structured and Modular programming. Basic Algorithms and problem solving, development of basic algorithms, analysing problem, designing solution, testing designed solution. Fundamental programming constructs, translation of algorithms to programmes, data types, control structures, functions, arrays, records, files, testing programmes.	5%
2.	**OBJECT ORIENTED PARADIGM:** Evolution of Object Oriented (OO) programming, OO concepts and principles, problem solving in OO paradigm, OO programme design process, classes, methods, objects and encapsulation; constructors and destructors, operator and function overloading, virtual functions, derived classes, inheritance and polymorphism. I/O and file processing, exception handling.	5%
3.	**DISCRETE STRUCTURES:** Introduction to logic and proofs: Direct proofs; proof by contradiction, Sets, Combinatorics, Sequences, Formal logic, Prepositional and predicate calculus, Methods of Proof, Mathematical Induction and Recursion, loop invariants, Relations and functions, Pigeonhole principle, Trees and Graphs, Elementary number theory, Optimization and matching. Fundamental structures: Functions; relations (more specifically recursions); pigeonhole principle; cardinality and countability, probabilistic methods.	5%
4.	**DATA STRUCTURES AND ALGORITHMS:** Introduction to data structures; Arrays, Stacks, Queues, Priority Queues, Linked Lists, Trees, Spanning Trees, Graphs	5%

	and Traversals. Recursion, sorting and searching algorithms, Shortest path algorithms, Hashing, Storage and retrieval properties and techniques for the various data structures. Algorithm Complexity, Polynomial and Intractable Algorithms, Classes of Efficient Algorithms, Divide and Conquer, Dynamic, Greedy	
5.	**DIGITAL LOGIC AND COMPUTER ORGANIZATION:** **5.1 Digital Logic__________________________2%** Overview of Binary Numbers, Boolean Algebra, switching algebra, and logic gates, Karnaugh Map and Quin-McCluskey methods, simplification of Boolean functions, Combinational Design; two level NAND/NOR implementation, Tabular Minimization, Combinational Logic Design: adders, subtracters, code converters, parity checkers, multilevel NAND/NOR/XOR circuits, MSI Components, design and use of encoders, decoders, multiplexers, BCD adders, and comparators, Latches and flip-flops, Synchronous sequential circuit design and analysis, Registers, synchronous and asynchronous counters, and memories, Control Logic Design. **5.2 Computer Organiztion_____________________3%** Fundamentals of Computer Designincluding performance measurements & quantitative principles, principles of Instruction Set Design, Operands, addressing modes and encoding, pipelining of Processors: Issues and Hurdles, exception handling features, Instruction-Level Parallelism and Dynamic handling of Exceptions, Memory Hierarchy Design, Cache Design, Performance Issues and improvements, Main Memory Performance Issues, Storage Systems, Multiprocessors and Thread Level Parallelism.	**5%**
6.	**OPERATING SYSTEMS:** History and Goals, Evolution of multi-user systems, Process and CPU management, Multithreading, Kernel and User Modes, Protection, Problems of cooperative processes, Synchronization, Deadlocks, Memory management and virtual memory, Relocation, External Fragmentation, Paging and Demand Paging, Secondary storage, Security and Protection, File systems, I/O systems, Introduction to distributed operating systems. Scheduling and dispatch, Introduction to concurrency.	**5%**

7.	**DATABASE SYSTEMS:** Basic database concepts; Entity Relationship modelling, Relational data model and algebra, Structured Query language; RDBMS; Database design, functional dependencies and normal forms; Transaction processing and optimization concepts; concurrency control and recovery techniques; Database security and authorization. Physical database design: Storage and file structure; indexed files; b-trees; files with dense index; files with variable length records; database efficiency and tuning.	5%
8.	**SOFTWARE ENGINEERING AND DEVELOPMENT:** Introduction to Computer-based System Engineering; Project Management; Software Specification; Requirements Engineering, System Modelling; Requirements Specifications; Software Prototyping; Software Design: Architectural Design, Object-Oriented Design, UML modelling, Function-Oriented Design, User Interface Design; Processes & Configuration Management; Introduction to advanced issues: Reusability, Patterns; Assignments and projects on various stages and deliverables of SDLC.	5%
9.	**COMPUTER COMMUNICATION AND NETWORKS:** Analogue and digital Transmission, Noise, Media, Encoding, Asynchronous and Synchronous transmission, Protocol design issues. Network system architectures (OSI, TCP/IP), Error Control, Flow Control, Data Link Protocols (HDLC, PPP). Local Area Networks and MAC Layer protocols (Ethernet, Token ring), Multiplexing, Switched and IP Networks, Inter-networking, Routing, Bridging, Transport layer protocols TCP/IP, UDP. Network security issues. Programming exercises, labs or projects involving implementation of protocols at different layers.	5%
10.	**COMPUTER ARCHITECTURE AND ASSEMBLY LANUGUAGE:** Microprocessor Bus Structure: Addressing, Data and Control, Memory Organization and Structure (Segmented and Linear Models), Introduction to Registers and Flags,	5%

	Data Movement, Arithmetic and Logic, Programme Control, Subroutines, Stack and its operation, Peripheral Control Interrupts. Objectives and Perspectives of Assembly Language, Addressing Modes, Introduction to the Assembler and Debugger, Manipulate and translate machine and assembly code. Interfacing with high level languages, Real-time application.	
11.	**THEORY OF AUTOMATA AND FORMAL LANGUAGES:** Finite State Models: Language definitions preliminaries, Regular expressions/Regular languages, Finite automata (FAs), Transition graphs (TGs), NFAs, Kleene's theorem, Transducers (automata with output), Pumping lemma and non regular language Grammars and PDA: Context free grammars, Derivations, derivation trees and ambiguity, Simplifying CFLs , Normal form grammars and parsing, Decidability, Chomsky's hierarchy of grammars Turing Machines Theory: Turing machines, Post machine, Variations on TM, TM encoding, Universal Turing Machine, Context sensitive Grammars, Defining Computers by TMs.	**5%**
12.	**ANALYSIS OF ALGORITHMS:** Asymptotic notations; Recursion and recurrence relations; Divide-and-conquer approach; Sorting; Search trees; Heaps; Hashing; Greedy approach; Dynamic programming; Graph algorithms; Shortest paths; Network flow; Disjoint Sets; Polynomial and matrix calculations; String matching; NP complete problems; Approximation algorithms	**5%**
13.	**ARTIFICIAL INTELLIGENCE:** Artificial Intelligence: Introduction, Intelligent Agents. Problem-solving: Solving Problems by Searching, Informed Search and Exploration, Constraint Satisfaction Problems, Adversarial Search. Knowledge and reasoning: Logical Agents, First-Order Logic, Inference in First-Order Logic, Knowledge Representation. Planning and Acting in the Real World. Uncertain knowledge and reasoning: Uncertainty, Probabilistic Reasoning, Probabilistic Reasoning over Time, Making Simple Decisions, Making Complex Decisions. Learning: Learning from Observations, Knowledge in Learning, Statistical Learning Methods,	**5%**

	Reinforcement Learning. Communicating, perceiving, and acting: Communication, Probabilistic Language Processing, Perception and Robotics. Introduction to LISP/PROLOG and Expert Systems (ES) and Applications.	
14.	**SYSTEM PROGRAMMING:** System Programming overview: Application Vs. System Programming, System Software, Operating System, Device Drivers, OS Calls. Window System Programming for Intel386 Architecture: 16 bit Vs 32 bit, Programming, 32 bit Flat memory model, Windows Architecture. Virtual Machine (VM)Basics, System Virtual Machine, Portable Executable Format, Ring O Computer, Linear Executable format, Virtual Device Driver (V + D), New Executable format, Module Management, COFF obj format 16 bit. (Unix) other 32-bit O.S Programming for I 386; Unix Binaryble format (ELF), Dynamic shared objects, Unix Kernel Programming (Ring O), Unix Device Architecture (Character & Block Devices), Device Driver Development, Enhancing Unix Kernel.	**5%**
15.	**NUMERICAL COMPUTING:** The concepts of efficiency, reliability and accuracy of a method.Minimising computational errors.Theory of Differences, Difference Operators, Difference Tables, Forward Differences, Backward Differences and Central Differences. Mathematical Preliminaries, Solution of Equations in one variable, Interpolation and Polynomial Approximation, Numerical Differentiation and Numerical Integration, Initial Value Problems for Ordinary Differential Equations, Direct Methods for Solving Linear Systems, Iterative Techniques in Matrix Algebra, Solution of non-linear equations.	**5%**
16.	**Quality Assurance:** The concepts of software quality assurance ant testing methodologies/techniques for both black box testing & white box testing and related tools. Major testing techniques should be covered e.g. Unit testing, integration testing, regression testing, data base testing, User Acceptance Testing (UAT), smoke testing, Load testing, Performance Testing alongwith their sub-techniques like branch testing, data definition testing, boundary value testing etc.	**25%**
	Total	**100%**

1. **Director Software Development**
2. **Principal Software Engineer**
3. **Principal Computer Engineer**
4. **Principal Business Analyst**

Sr. No.	Core Areas	Percentage
1.	Programming Fundamentals	06%
2.	Object Oriented Paradigm	05%
3.	Discrete Structures	05%
4.	Data Structures and Algorithms	09%
5.	Digital Logic and Computer Organization	06%
6.	Operating Systems	06%
7.	Database Systems	07%
8.	Software Engineering & Development	05%
9.	Computer Communication and Networks	06%
10.	Computer Architecture & Assembly Language	08%
11.	Theory of Automata and Formal Languages	10%
12.	Analysis of Algorithms	10%
13.	Artificial Intelligence	07%
14.	System Programming	05%
15.	Numerical Computing	05%
	Total	**100%**

Computer Science (Detailed)		
Sr. No.	**Core Areas**	**Percentage**
1.	**PROGRAMMING FUNDAMENTALS:** Overview of computers and programming. Overview of language. Basics of structured and Modular programming. Basic Algorithms and problem solving, development of basic algorithms, analysing problem, designing solution, testing designed solution. Fundamental programming constructs, translation of algorithms to programmes, data types, control structures, functions, arrays, records, files, testing programmes.	**6%**
2.	**OBJECT ORIENTED PARADIGM:** Evolution of Object Oriented (OO) programming, OO concepts and principles, problem solving in OO paradigm, OO programme design process, classes, methods, objects and encapsulation; constructors and destructors, operator and function overloading, virtual functions, derived classes, inheritance and polymorphism. I/O and file processing, exception handling.	**5%**
3.	**DISCRETE STRUCTURES:** Introduction to logic and proofs: Direct proofs; proof by contradiction, Sets, Combinatronics, Sequences, Formal logic, Prepositional and predicate calculus, Methods of Proof, Mathematical Induction and Recursion, loop invariants, Relations and functions, Pigeonhole principle, Trees and Graphs, Elementary number theory, Optimization and matching. Fundamental structures: Functions; relations (more specifically recursions); pigeonhole principle; cardinality and countability, probabilistic methods.	**5%**
4.	**DATA STRUCTURES AND ALGORITHMS:** Introduction to data structures; Arrays, Stacks, Queues, Priority Queues, Linked Lists, Trees, Spanning Trees, Graphs	**9%**

	and Traversals. Recursion, sorting and searching algorithms, shortest path algorithms, Hashing, Storage and retrieval properties and techniques for the various data structures. Algorithm Complexity, Polynomial and Intractable Algorithms, Classes of Efficient Algorithms, Divide and Conquer, Dynamic, Greedy	
5.	**DIGITAL LOGIC AND COMPUTER ORGANIZATION:** **5.1 Digital Logic** ________ **3%** Overview of Binary Numbers, Boolean Algebra, switching algebra, and logic gates, Karnaugh Map and Quin-McCluskey methods, simplification of Boolean functions, Combinational Design; two level NAND/NOR implementation, Tabular Minimization, Combinational Logic Design: adders, subtracters, code converters, parity checkers, multilevel NAND/NOR/XOR circuits, MSI Components, design and use of encoders, decoders, multiplexers, BCD adders, and comparators, Latches and flip-flops, Synchronous sequential circuit design and analysis, Registers, synchronous and asynchronous counters, and memories, Control Logic Design. **5.2 Computer Organiztion** ________ **3%** Fundamentals of Computer Design including performance measurements & quantitative principles, principles of Instruction Set Design, Operands, addressing modes and encoding, pipelining of Processors: Issues and Hurdles, exception handling features, Instruction-Level Parallelism and Dynamic handling of Exceptions, Memory Hierarchy Design, Cache Design, Performance Issues and improvements, Main Memory Performance Issues, Storage Systems, Multiprocessors and Thread Level Parallelism.	**6%**
6.	**OPERATING SYSTEMS:** History and Goals, Evolution of multi-user systems, Process and CPU management, Multithreading, Kernel and User Modes, Protection, Problems of cooperative processes, Synchronization, Deadlocks, Memory management and virtual memory, Relocation, External Fragmentation, Paging and Demand Paging, Secondary storage, Security and Protection, File systems, I/O systems, Introduction to distributed operating systems. Scheduling and dispatch, Introduction to concurrency.	**6%**

7.	**DATABASE SYSTEMS:** Basic database concepts; Entity Relationship modelling, Relational data model and algebra, Structured Query language; RDBMS; Database design, functional dependencies and normal forms; Transaction processing and optimization concepts; concurrency control and recovery techniques; Database security and authorization. Physical database design: Storage and file structure; indexed files; b-trees; files with dense index; files with variable length records; database efficiency and tuning.	7%
8.	**SOFTWARE ENGINEERING AND DEVELOPMENT:** Introduction to Computer-based System Engineering; Project Management; Software Specification; Requirements Engineering, System Modelling; Requirements Specifications; Software Prototyping; Software Design: Architectural Design, Object-Oriented Design, UML modelling, Function-Oriented Design, User Interface Design; Quality Assurance; Processes & Configuration Management; Introduction to advanced issues: Reusability, Patterns; Assignments and projects on various stages and deliverables of SDLC.	5%
9.	**COMPUTER COMMUNICATION AND NETWORKS:** Analogue and digital Transmission, Noise, Media, Encoding, Asynchronous and Synchronous transmission, Protocol design issues. Network system architectures (OSI, TCP/IP), Error Control, Flow Control, Data Link Protocols (HDLC, PPP). Local Area Networks and MAC Layer protocols (Ethernet, Token ring), Multiplexing, Switched and IP Networks, Inter-networking, Routing, Bridging, Transport layer protocols TCP/IP, UDP. Network security issues. Programming exercises, labs or projects involving implementation of protocols at different layers.	6%
10.	**COMPUTER ARCHITECTURE AND ASSEMBLY LANUGUAGE:** Microprocessor Bus Structure: Addressing, Data and Control, Memory Organization and Structure (Segmented and Linear Models), Introduction to Registers and Flags, Data Movement, Arithmetic and Logic, Programme Control, Subroutines, Stack and its operation, Peripheral	8%

	Control Interrupts. Objectives and Perspectives of Assembly Language, Addressing Modes, Introduction to the Assembler and Debugger, Manipulate and translate machine and assembly code. Interfacing with high level languages, Real-time application.	
11.	**THEORY OF AUTOMATA AND FORMAL LANGUAGES:** Finite State Models: Language definitions preliminaries, Regular expressions/Regular languages, Finite automata (FAs), Transition graphs (TGs), NFAs, Kleene's theorem, Transducers (automata with output), Pumping lemma and non-regular language Grammars and PDA: Context free grammars, Derivations, derivation trees and ambiguity, Simplifying CFLs , Normal form grammars and parsing, Decidability, Chomsky's hierarchy of grammars Turing Machines Theory: Turing machines, Post machine, Variations on TM, TM encoding, Universal Turing Machine, Context sensitive Grammars, Defining Computers by TMs.	10%
12.	**ANALYSIS OF ALGORITHMS:** Asymptotic notations; Recursion and recurrence relations; Divide-and-conquer approach; Sorting; Search trees; Heaps; Hashing; Greedy approach; Dynamic programming; Graph algorithms; Shortest paths; Network flow; Disjoint Sets; Polynomial and matrix calculations; String matching; NP complete problems; Approximation algorithms	10%
13.	**ARTIFICIAL INTELLIGENCE:** Artificial Intelligence: Introduction, Intelligent Agents. Problem-solving: Solving Problems by Searching, Informed Search and Exploration, Constraint Satisfaction Problems, Adversarial Search. Knowledge and reasoning: Logical Agents, First-Order Logic, Inference in First-Order Logic, Knowledge Representation. Planning and Acting in the Real World. Uncertain knowledge and reasoning: Uncertainty, Probabilistic Reasoning, Probabilistic Reasoning over Time, Making Simple Decisions, Making Complex Decisions. Learning: Learning from Observations, Knowledge in Learning, Statistical Learning Methods, Reinforcement Learning. Communicating, perceiving, and acting: Communication, Probabilistic Language	7%

	Processing, Perception and Robotics. Introduction to LISP/PROLOG and Expert Systems (ES) and Applications.	
14.	**SYSTEM PROGRAMMING:** System Programming overview: Application Vs. System Programming, System Software, Operating System, Device Drivers, OS Calls. Window System Programming for Intel386 Architecture: 16 bit Vs 32 bit, Programming, 32 bit Flat memory model, Windows Architecture. Virtual Machine (VM)Basics, System Virtual Machine, Portable Executable Format, Ring O Computer, Linear Executable format, Virtual Device Driver (V + D), New Executable format, Module Management, COFF obj format 16 bit. (Unix) other 32-bit O.S Programming for I 386; Unix Binaryble format (ELF), Dynamic shared objects, Unix Kernel Programming (Ring O), Unix Device Architecture (Character & Block Devices), Device Driver Development, Enhancing Unix Kernel.	5%
15.	**NUMERICAL COMPUTING:** The concepts of efficiency, reliability and accuracy of a method. Minimising computational errors. Theory of Differences, Difference Operators, Difference Tables, Forward Differences, Backward Differences and Central Differences. Mathematical Preliminaries, Solution of Equations in one variable, Interpolation and Polynomial Approximation, Numerical Differentiation and Numerical Integration, Initial Value Problems for Ordinary Differential Equations, Direct Methods for Solving Linear Systems, Iterative Techniques in Matrix Algebra, Solution of non-linear equations.	5%
	Total	**100%**