THIRDYEAR :COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER - V

Subject Code	Nomenclature of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	ThDuration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 5.1	Data Communication	3	0	0	3	100	25	--	--	--	**125**
COMP 5.2	Automata Languages and Computation	3	1	2	3	100	25	25		--	**150**
COMP 5.3	Cryptography and Coding Theory	3	1	0	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**	**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.

THIRDYEAR: COMPUTER ENGINEERING

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER - VI

Subject Code	Nomenclature of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	ThDuration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 6.1	Software Testing and Quality Assurance	3	0	0	3	100	25	--	--	--	**125**
COMP 6.2	Design and Analysis of Algorithms	3	1	0	3	100	25	--	--	--	**125**
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	Embedded System Design	3	1	2	3	100	25	25	--	--	**150**
COMP 6.6	Computer Networks	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.

APPENDIX

COMP 5.1 DATA COMMUNICATION

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 hrs	100	25	-	-	-	**125**

Course Objectives:

1. To focus on imparting knowledge about various components of data communications emphasizing on the physical layer and data link layer of the OSI stack.
2. To understand the conceptual and analytical differences between analog and digital communication.

Course Outcomes:

The student after undergoing this course will be able to:

1. Understand the basic concepts of data communication components used at various transmission speeds.
2. Identify the characteristics and analyze specific role of Data Communication technologies such as ATM, wireless, satellite and fiber optic communication.

<u>UNIT - 1</u> (12 Hours)

Introduction to Data Communication, Networks, Protoc. ols and Standards. Network Models: Layered Task, The OSI Reference Model, TCP/IP protocol Suite, Addressing
Data and Signals: Analog and Digital, Periodic analog signals, digital signal, transmission impairment, data rate limits, performance. Digital Transmission: digital-to digital conversion, analog-to-digital conversion transmission modes

<u>UNIT - 2</u> (12 Hours)

Analog Transmission: Digital-to- analog conversion, analog-to-analog conversion. Multiplexing and Spread Spectrum. Transmission Media: Guided Media, Unguided Media
Switching: Circuit Switched Networks, Datagram Networks. Telephone Network: Dial-up modems, DSL, Cable TV networks

UNIT - 3 (12 Hours)

Error Detection and Correction: , Block Coding, Linear Block Codes, Cyclic Block Codes,
Data Link Control: Framing, Flow and Error Control, Protocols, Noiseless channels, noisy channels, HDLC .Random Access. Channelization.

UNIT - 4 (12 Hours)

Connecting LAN's, Backbone Networks and Virtual LAN's. Wireless WAN's: Cellular Telephony, satellite Networks. Virtual- Circuit Networks: Frame Relay, ATM, ATM LAN

Recommended Readings:

1. B.A. Forouzan; Data Communication and Networking; Tata McGraw Hill, 4th Edition
2. Andrew S. Tanenbaum ;Computer Networks; Pearson Education
3. William Stallings; Data and Computer Communication; Seventh edition

COMP5.2 AUTOMATA LANGUAGE AND COMPUTATION

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
						Marks					
		L	T	P	Th Duration(Hrs)	Th	S	TW	P	O	Total
COMP5.2	Automata language and Computation	3	0	2	3	100	25	--	--	--	**125**

Course Objectives:

1. To introduce concepts in automata theory and theory of computation.
2. To learn different formal language classes and their relationships.
3. To design grammars and recognizers for different formal languages.
4. To study the decidability and intractability of computation problems.

Course Outcomes:

The student after undergoing this course will be able to:

1. Identify formal language classes and Explain the properties of languages, grammars and automata.
2. Apply the techniques to transform between equivalent deterministic and non-deterministic finite automata and regular expressions.
3. Design grammars and automata (recognizers) for different language classes.
4. Perform the Simplification of automata and Context free grammars.
5. Explain the concepts of context-free languages, pushdown automata and Turing recognizable languages.

UNIT – 1 **(12 Hours)**

Mathematical Preliminaries and Notation: Sets, Functions and Relations, Graphs and Trees, Proof Techniques. Languages, Grammars and Automata. A Hierarchy of Formal Languages and Automata, Recursive and Recursively Enumerable Languages, Unrestricted Grammars, Context-Sensitive Grammars and Languages, Closure properties of Regular and Context Free languages. The Chomsky Hierarchy. Finite Automata: Deterministic Finite Accepters, Deterministic Accepters and Transition Graphs.

UNIT - 2 **(12 Hours)**

Nondeterministic Finite Accepters, Equivalence of Deterministic and Nondeterministic Finite Accepters, Reduction of the Number of States in Finite Automata, Regular Expressions: Connection Between Regular Expressions and Regular Languages.

Right- and Left-Linear Grammars. Equivalence of Regular Languages and Regular Grammars. A Pumping Lemma for regular languages.

UNIT - 3 (12 Hours)

Context-Free Grammars, Leftmost and Rightmost Derivations, Derivation Trees, Parsing and Ambiguity. Methods for Transforming Grammars: Substitution Rule, Removing Useless Productions, lambda-Productions, and Unit-Productions. Chomsky Normal Form, and Greibach Normal Form. Deterministic and Nondeterministic Pushdown Automata, Pushdown Automata for Context-Free Languages, Context-Free Grammars for Pushdown Automata. A Pumping Lemma for Context-Free Languages.

UNIT - 4 (12 Hours)

Definition of a Turing Machine, Turing Machines as Language Accepters, Turing Machines as Transducers, Combining Turing Machines for Complicated Tasks, Turing's Thesis.. Turing Machines with More Complex Storage, Multitape Turing Machines, Multidimensional Turing Machines. Nondeterministic Turing Machines, A Universal Turing Machine, Linear Bounded Automata. Computability and Decidability: Turing Machine Halting Problem.

.

Recommended Readings:

1. Peter Linz; An introduction to Formal Languages and Automata; Jones & Bartlett Learning, 2006
2. K.L.P Mishra, N. Chandrasekaran; Theory of Computer Science – Automata, languages and Computation; PHI Publications; Third Edition ; 2008.
3. John C Martin; Introduction to languages and the theory of computation; Tata McGraw Hill, Fourth Edition, 2010.
4. John E. Hopcraft and Jeffery D. Ullman; Introduction to Automata Theory, Languages and Computation; Narosa Publishing House.
5. Michael Sipser; Introduction to Theory of Computation; PWS Publishing Company.
6. A.A Puntambekar; Formal Languages and Automata Theory; Technical Publications Pune;

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Design and implementation of Non-Deterministic Finite Automata (NFA).
2. Design and implementation of Deterministic Finite Automata (DFA).
3. Transform the equivalent DFA from the Non-Deterministic Finite Automata.
4. Obtain the minimized DFA from the given DFA.
5. Convert the given RE into Finite Automata.
6. Design and Implementation of Mealy machine.
7. Design and Implementation of Moore machine.
8. Prove Pumping lemma for regular languages.
9. Prove Pumping lemma for context-free languages.
10. Design and Implementation of Pushdown Automata.
11. Design and Implementation of Turing Machine.
12. Convert the grammar to its equivalent automata and vice versa.

COMP5.3 CRYPTOGRAPHY AND CODING THEORY

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
COMP5.3	Cryptography and Coding Theory	3	1	0	3	100	25	--	--	--	**125**

Course Objectives:

1. To familiarize students with the theory and practice of the coding and encryption schemes.
2. To learn fundamental properties, most relevant examples and most important applications of Block Codes.
3. To understand various techniques for symmetric and asymmetric cryptography.
4. To familiarize the techniques for hashing and digital signature.

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the foundations of number theory and its applications in building crypto systems.
2. Analyze which error-correction coding scheme is most appropriate for a given demand.
3. Explain the relations existing among different areas of mathematics, especially algebra, coding theory and the theory of self -correcting codes.
4. Describe which encryption techniques are currently used in the different digital systems and to how they work.

<u>UNIT – 1</u> (12 Hours)

Basic Number Theory: Basic Notions, Congruences, The Chinese Remainder Theorem, Modular Exponentiation, Fermat and Euler, Primitive Roots,

Classical Cryptosystems: Shift Ciphers, Affine Ciphers ,TheVigenere Cipher, Substitution Ciphers , Sherlock Holmes , The Playfair and ADFGX Ciphers One-Time Pads, Pseudo-random Bit Generation, LFSR Sequences, Enigma.

UNIT - 2 (12 Hours)

The Data Encryption Standard: A Simplified DES-Type Algorithm, Differential Cryptanalysis, DES, Modes of Operation, Breaking DES, Meet-in-the-Middle Attacks Password Security. The Advanced Encryption Standard: Rijndael Algorithm, The Layers, Decryption, Design Considerations.

UNIT - 3 (12 Hours)

The RSA Algorithm, Attacks on RSA, Primality Testing, Factoring, The RSA Challenge, The Public Key Concept, Discrete Logarithms, Computing Discrete Logs, Bit Commitment Diffie-Hellman Key Exchange, The El Gamal Public Key Cryptosystem.

UNIT - 4 (12 Hours)

Hash Functions: A Simple Hash Example, The Secure Hash Algorithm, Birthday Attacks, Digital Signatures, RSA Signatures, The El Gamal Signature Scheme, Hashing and Signing..

Error Correcting Codes, Bounds on General Codes, Linear Codes, Hamming Codes, Golay Codes, Cyclic Codes, BCH Codes and Reed-Solomon Codes

Recommended Readings:

1) Introduction to Cryptography with Coding Theory, 2nd edition, Wade Trappe and Lawrence C. Washington, Pearson Education, 2011
2) Lin, S.; Costello, D.J.. Error control coding: fundamentals and applications. Prentice-Hall, 2004.
3) R. Hill ; A first course in Coding Theory,; Oxford, 0-19-8538030
4) J. A. Buchmann; Introduction to Cryptography; Springer, 0387950346
5) W. C. Huffman & V. Pless; Fundamentals of Error-Correcting Codes; Cambridge, 521782805
6) S. Roman; Introduction to Coding and Information Theory,; Springer
7) A. Menezes, P. van Oorschot, and S. Vanstone; Handbook of Applied Cryptography; CRC
8) N. Ferguson & B. Schneier; Practical Cryptography; John Wiley
9) W. Mao; Modern Cryptography Theory and Practice; Prentice Hall,
10) H. Niederreiter editor; Coding Theory and Cryptology; World Scientific/Singapur U. Press

COMP 5.4 COMPUTER HARDWARE 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 5.4	Computer Hardware Design	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. To understand the basic principles of Digital System design.
2. To use hardware description language, VHDL, in the design process.
3. To automatically synthesize digital hardware from VHDL description using CAD tools.
4. To equip students with skills to design systems at high level and express algorithms in VHDL.

Course Outcomes:

The student after undergoing this course will be able to:

1. Relate the constructs of VHDL to corresponding hardware
2. Design digital logic circuits using Computer Aided Tools.
3. Model, Simulate and Synthesize digital systems using VHDL.
4. Design and prototype with programmable logic.
5. Test a design described in VHDL using a test bench written in VHDL.

UNIT - 1 (12 Hours)

Introduction to VHDL: Computer-Aided Design, Hardware Description Languages, VHDL Description of Combinational Circuits, VHDL Modules, Sequential Statements and VHDL Processes, Modeling Flip-Flops Using VHDL Processes, Processes Using Wait Statements, Two Types of VHDL Delays: Transport and Inertial Delays, Compilation, Simulation, and Synthesis of VHDL Code, VHDL Data Types and Operators, Simple Synthesis Examples, VHDL Models for Multiplexers, VHDL Libraries, Modeling Registers and Counters Using VHDL Processes, Behavioral and

Structural VHDL, Variables, Signals, and Constants, Arrays, Loops in VHDL, Assert and Report Statements. Brief Overview of Programmable Logic Devices, Simple Programmable Logic Devices (SPLDs), Complex Programmable Logic Devices (CPLDs), Field-Programmable Gate Arrays (FPGAs)

UNIT - 2 (12 Hours)

BCD to 7-Segment Display Decoder, A BCD Adder, 32-Bit Adders, Traffic Light Controller, State Graphs for Control Circuits, Scoreboard and Controller, Synchronization and Debouncing, A Shift-and-Add Multiplier, Array Multiplier, A Signed Integer,Fraction Multiplier, Keypad Scanner, Binary Dividers. State Machine Charts, Derivation of SM Charts, realization of SM Charts, Implementation of the Dice Game, Microprogramming, Linked State Machines.

UNIT - 3 (12 Hours)

Implementing Functions in FPGAs, Implementing Functions Using Shannon's Decomposition, Carry Chains in FPGAs, Cascade Chains in FPGAs, Examples of Logic Blocks in Commercial FPGAs, Dedicated Memory in FPGAs, Dedicated Multipliers in FPGAs, Cost of Programmability, FPGAs and One-Hot State Assignment, FPGA Capacity: Maximum Gates Versus Usable Gates, Design Translation (Synthesis), Mapping, Placement, and Routing. Representation of Floating-Point Numbers, Floating-Point Multiplication, Floating-Point Addition, Other Floating-Point Operations

UNIT - 4 (12 Hours)

VHDL Functions, VHDL Procedures, Attributes, Creating Overloaded Operators, Multi-Valued Logic and Signal Resolution,The IEEE 9-Valued Logic System, SRAM Model Using IEEE 1164, Model for SRAM Read,Write System, Generics, Named Association, Generate Statements, Files and TEXTIO. Testing Combinational Logic , Testing Sequential Logic, Scan Testing, Boundary Scan, Built-In-Self-Test.

Recommended Readings:

1. Charles Roth; Digital Systems Design Using VHDL; Cengage Learning ,2006
2. Charles Roth ,Lizy John; Digital Systems Design Using VHDL; 2nd Ed., Thomson, 2008.
3. Volnei A. Pedroni; Circuit Design with VHDL; MIT Press, 2004
4. Stephen Brown, Z. Vranesic; Fundamentals of Digital Logic Design with VHDL Second Edition; McGrawHill.2007
5. Jayaram Bhasker; VHDL Primer, A; 3rd Edition; Prentice Hall; 1999

List of Experiments in Computer Hardware Design:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Introduction to lab hardware & software (Tutorial)
2. Combinational Logic Design Using Schematic Capture
3. Implementation of decoders.
4. Implementation of adders and subtractors
5. Sequential Logic Design Using Schematic Capture
6. Combinational Logic Design Using VHDL
7. Sequential Logic Design Using VHDL
8. Parameterized VHDL Universal Register/Counter
9. Implementation of interfacing with stepper motor, seven segment displays etc.

COMP 5.5 DATABASE MANAGEMENT SYSTEMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 5.5	Database Management System	3	1	2	3	100	25	--	25		**150**

Course Objectives:

1. To understand the basic concepts and the applications of database systems.
2. To master the basics of SQL and construct queries using SQL.
3. To understand the relational database design principles.
4. To become familiar with the basic issues of transaction processing and concurrency control.

Course Outcomes:

The student after undergoing this course will be able to:

1. Demonstrate the basic elements of a relational database managament system.
2. Ability to identify the data models for relevant problems.
3. Ability to design entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respect data.
4. Apply normalization for the development of application softwares.

UNIT -1 (12 Hours)

Introduction to Databases: Characteristic of Database Approach, Advantages of using DBMS approach. Overview of Database Languages and Architecture: Data Models, Schemas and instances, Three Scheme Architecture and Data Independence. Relational model concepts, Constraints and relational Database schema , Update operation, transactions and dealing with constraint violation. Formal Relational languages: Unary relational operations, relational algebra operations from set theory, Binary relational operations JOIN and DIVISION. Additional relational operations.

UNIT - 2 (12 Hours)

SQL data Definition and Data types, specifying constraints in SQL, Basic retrieval in SQL. INSERT,DELETE and UPDATE statement in SQL. More Complex SQL retrieval queries. Specifying constraints as Assertions and Actions as triggers. VIEWS in SQL. Schema Change statement.

UNIT – 3 (12 Hours)

Using High level conceptual Data model for Database Design, Entity type, entity set, attributes and keys, relationship type, relationship set roles and structural constraints, weak entity type, ER diagrams, Naming conventions and Design issues, subclasses, super classes and Inheritance, Specialization and Generalization in EER. Mapping the conceptual design into logical design: ER to relational mapping. Introduction to Normalization. Informal design guidelines for relational schemas, Functional dependencies, Normal forms based on primary keys(1NF,2NF ,3NF),BCNF. Database Design Theory: Inference rules, Equivalence and minimal cover.

UNIT - 4 (12 Hours)

Foundation of Database Transaction Processing: Transaction and system concepts, desirable properties of transaction, Characterizing schedules based on recoverability Characterizing schedules based on serializability Introduction to protocols for concurrency control in Databases: Two phase locking technique for concurrency control , concurrency control based on timestamp ordering, Multiversion concurrency control technique, validation concurrency control technique. Introduction to Database security: Security Issues

Recommended Readings:

1. Ramez Elmasri, Shamkat B. Navathe; Database Systems; 6th edition; Pearson Education; 2013.
2. A.Silberschatz, H.F. Korth, S.Sudarshan; Data base System Concepts; McGraw Hill; VI edition, 2006.
3. Raghurama Krishnan, Johannes Gehrke; Data base Management Systems; TATA McGrawHill; 3rd Edition; 2003.
4. Thomas Connolly, Carolyn Begg; Database Systems; A Practical approach to Design implementation and Management Fourth edition; Pearson education.
5. P.K. Das Gupta; Database Management System Oracle SQL and PL/SQL; PHI.
6. C.J. Date; Introduction to Database Systems; Pearson Education.

List of Experiments in Database Management Systems:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. To study Data Definition language Statements.
2. To study Data Manipulation Statatements.
3. Study of SELECT command with different clauses
4. Study of GROUP functions (avg, count, max, min, Sum).
5. Study of various type of SET OPERATORS (Union, Intersect, Minus).
6. Study of various type of Integrity Constraints.
7. Study of Various type of JOINS.
8. To study Views
9. To Study Triggers

COMP 5.6 OPERATING SYSTEMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 5.6	Operating Systems	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

1. To provide students a comprehensive introduction to understand the underlying principles, techniques and approaches which constitute a coherent body of knowledge in operating systems.
2. To provide students an understanding of concurrent processes, multi-threads, CPU scheduling, memory management, file system, storage subsystem, and input/output management.
3. To provide students knowledge of the components and management aspects of different types of operating systems.
4. To equip students with necessary skills required for Shell Programming.

Course Outcomes:

The student after undergoing this course will be able to:

1. Understand the fundamental concepts of operating systems, its evolution and various architectures.
2. Understand relationship between subsystems of a modern operating system.
3. Gain knowledge on processes, threads, concurrency control, synchronization, CPU scheduling and semaphores.
4. Analyze algorithms used in process management, deadlock handling, memory management, file systems and I/O systems.
5. Be familiar with various types of operating systems including UNIX.

<u>UNIT -1</u> (12 Hours)

Process management: Processes concept, Process scheduling, Operations on processes, Interprocess communication. Threads: Overview, Multithreading models, Threading issues. CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms: FCFS, SJF, SRTF / SRTN, Priority Scheduling ,Round Robin Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling, Fair Share Scheduling, Multiprocessor Scheduling, Real – Time Scheduling

UNIT - 2 (12 Hours)

Process Synchronization: Critical Section Problem, Petersons solution, Synchronisation hardware support, Mutex locks, Semaphores, Classical problems of syncronisation using semaphores (Producer – Consumer problem, Readers – Writers problem,Dining philosophers Problem), Monitors.

Deadlocks: System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.

UNIT – 3 (12 Hours)

Memory Management: Background, Swapping, Contiguous allocation, Segmentation, Paging, Structure of the page table

Virtual Memory: Demand Paging, Copy on write, Page replacement algorithms (FIFO, Optimal page replacement, Least Recently used), Allocation of frames, Thrashing.

File System Interface: File Concept, Access methods, Directory and Disk Structure.

File system implementation: File system structure, Implementation, Directory implementation, Allocation methods

UNIT – 4 (12 Hours)

I/O Systems: I/O Hardware, Application I/O Interface, Kernel I/O subsystem, Transforming I/O requests to hardware operations.

Secondary Storage structure: Disk structure and attachment, Disk scheduling, Disk management

Linux Commands: Basic Linux commands, Essential Shell Programming.

Recommended Readings:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne; Operating System Concepts; 9th Edition
2. William Stallings; Operating systems internals and design principles;7th edition
3. Sumitabha Das; UNIX – Concepts and applications;4th edition
4. A.S Tanenbaum; Operating systems, Design and implementation;3rd edition
5. Milenkovic; Operating Systems,2nd edition

6. William E. Shotts, Jr; The Linux Command Line: A Complete Introduction;3rd edition

List of Experiments in Operating systems:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Study of various types of operating systems
2. Implementation of system calls
3. Non preemptive CPU scheduling algorithms
4. Preemptive CPU scheduling algorithms
5. Implementation of threads
6. Process synchronization using semaphores
7. Implementation of deadlock avoidance scheme
8. Memory allocation techniques
9. Page replacement methods
10. Disk scheduling algorithms
11. Shell scripting

COMP6.1 SOFTWARE TESTING AND QUALITY ASSURANCE

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	Software Testing and Quality Assurance	3	0	0	3	100	25	--	--	--	**125**

Course Objectives:

1. To develop and implement an effective testing strategy,.
2. To plan and prepare appropriate tests for all phases of software development
3. To measure and control the quality of the testing.
4. To understand the significance of finding and resolving errors early.
5. To develop strategies and techniques for building-in quality from the start.

Course Outcomes:

The student after undergoing this course will be able to:

1. Manage, plan and prepare rigorous, formal, visible and repeatable tests that will fully exercise software, in the development of quality systems.
2. Apply. different testing approaches to all stages of software development
3. Prepare test plans, strategy, specifications, procedures and controls to provide a structured approach to testing.
4. Apply the techniques and methods covered to testing packages.
5. Describe the different types of testing tools available and identify the appropriate types of tools for their needs.

UNIT -1 (12 Hours)

Quality perspective and expectations, Quality framework and ISO 9126, Correctness and defects. Quality Assurance: Classification, Defect prevention, Defect reduction, Defect containment. Quality Assurance in context: Handling discovered defects during QA activities, QA activities, Verification and validation perspective.

Quality Engineering: Activities & Process, Quality planning, Quality assessment & improving, ISO 9000 series standards. Capability Maturity Model integration for software engineering.

UNIT -2 (12 Hours)

Purpose, activities, process and context, issues and questions about testing, Functional v/s structural testing, Coverage based v/s usage based testing. Test planning and preparation, Test execution, result checking and measurement, Analysis and follow up, Activities, people and management. Coverage and usage testing based on checklists and partitions: Checklist based testing and limitations. Testing for partition coverage, Usage based statistical testing with Musa's operational profiles. Input domain partitioning and testing, simple domain analysis and extreme point combination strategies, Testing strategies based on boundary analysis.

UNIT - 3 **(12 Hours)**

Defect prevention and process improvement: Basic concepts and generic approaches, Root cause analysis for defect prevention, Training for defect prevention, Defect prevention techniques. Basic Control flow testing, Data Dependency and data flow testing. Finite State Machines (FSM)and testing, FSM testing, FSM based testing of web based applications, Markov Chains and Unified Markov Models (UMM) for testing, Using UMM's for usage based statistical testing, Testing based on Web usages.

UNIT - 4 **(12 Hours)**

Software testing tools and overview: Need for automated testing tools, Taxonomy of testing tools, Functional/Regression testing tools, Performance testing tools, Testing management tools, Source code testing tools, Selection of testing tools. Case study: Overview of WinRunner, Loadrunner, Quick Test Professional and SQA Robot.

Recommended Readings:

1. Software Quality Engineering – Testing, Quality Assurance and Quantifiable Improvement by Jeff Tian, Edition 2006, ISBN: 81-265-0805-1
2. Software Testing Tools by Dr. K.V.K.K. Prasad.
3. Effective methods for Software testing by William E. Perry, 3rd edition.
4. Introducing Software testing by Louise Tamares, ISBN: 81-7808-678-6

COMP 6.2 DESIGN AND ANALYSIS OF ALGORITHMS

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 6.2	Design and Analysis of Algorithms	3	1		3	100	25		-	-	**125**

Course Objectives:

1. To learn algorithm designing techniques and understand the asymptotic behavior of algorithms in general and sorting algorithms in particular.
2. To acquire the knowledge of analyzing algorithms using the Recurrence relation techniques.
3. To gain the complete understanding of Dynamic Programming, Greedy Algorithmic strategies, Backtracking methods and Branch and Bound Techniques.
4. To comprehend the analysis of graph algorithms and to learn about the single source shortest paths and all pairs shortest path algorithms and their analysis in graph theory.
5. To get an in-depth understanding of randomized algorithms, NP-Completeness and reduction and FFT and sorting networks.

Course Outcomes:

The student after undergoing this course will be able to:

1. Describe and analyze the complexity and asymptotic behavior of algorithms in general and sorting algorithms in particular.
2. Trace time complexity of algorithms using the Recurrence relation techniques.
3. Explain and apply dynamic programming, greedy algorithms, backtracking and branch and bound techniques to various problems.
4. Analyze graph algorithms and find the single source shortest paths and all pairs shortest paths in any graph.
5. Describe the classes P, NP, and NP-Complete and be able to prove that a certain problem is NP-Complete.

UNIT - 1 (12 Hours)

Introduction to Algorithms. Algorithm specification, Performance analysis, Growth of Function, Recurrences, Randomized Algorithms.
Divide and conquer: General Method, Binary Search, Finding the maximum and minimum, Merge Sort, Quick Sort, Selection Sort, Strassen's Matrix Multiplication.
Greedy method strategy: General method, Knapsack problem, Job sequencing with deadlines.

UNIT - 2 (12 Hours)

Dynamic Programming: General method, Multistage graphs, Optimal Binary Search Trees, 0/1 Knapsack, Travelling Salesperson problem, Flow Shop Scheduling.
Backtracking: General method, 8-Queens Problem, Sum of Subsets Problem, Graph Coloring, Hamiltonian Cycles, Knapsack Problem.
Branch and Bound: The Method,0/1 Knapsack Problem, Traveling Salesperson.

UNIT - 3 (12 Hours)

Internet Algorithms: Strings and patterns matching algorithm. Tries. Text compression. Text similarity testing.
Introduction to parallelism models: Simple algorithms for parallel computers, CRCW and EREW algorithms. Probabilistic Algorithms: Expected versus average time, Pseudorandom generation, Buffon's needle, numerical integration, Probabilistic counting, Monte Carlo algorithms .

UNIT - 4 (12 Hours)

NP-Hard and NP-Complete Problems: Basic concepts, NP-Hard Graph Problems: CDP, NCDP, CNDP, DHC NP-Hard Scheduling Problems: Scheduling Identical Processors, Flow-Shop Scheduling, Job Shop Scheduling. Approximation algorithms: Vertex cover problem, traveling-salesperson problem, set-covering problem.

Recommended Readings:

1. Computer Algorithms by Horowitz, Sartaj Sahni. Rajasekharan – Galgotia, ISBN: 9788175152571
2. Computer Algorithms – Saar Baase. PHI , ISBN: 0201612445
3. Fundamentals of Algorithms by Gilles Brassard and Paul Bratly. PHI, ISBN: 9780133350685.
4. Algorithm Design Foundation, Analysis and Internet Examples by Michael Goodrich & Roberto Tamassia,, Second Edition, Wiley student Edition.
5. Introduction to algorithms by Thomas H cormen, Charles E Leiserson, Ronald L Rivest. PHI, ISBN: 81-203-1353-4

COMP6.3 ARTIFICIAL INTELLIGENCE

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 6.3	Artificial Intelligence	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

1. To study the concepts of Artificial Intelligence
2. To learn the methods of solving problems using Artificial Intelligence
3. To know various AI search algorithms
4. To study the techniques nod machine learning, natural language processing, experts systems and planning.

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the concepts of A.I. problem solving.
2. Identify the knowledge representation schemes.
3. Apply the various reasoning mechanism in decision making.
4. Explain the concepts of planning, perception and learning and their applications.
5. Design an expert system for a given application domain.

UNIT -1 (12 Hours)

Introduction to Artificial Intelligence and A.I. Techniques. Problems, Problem Spaces & Search. Defining the Problem as state space search. Production Systems, Problem characteristics, Production System Characteristics, Design issues in Searching. Uninformed Search Techniques: BFS, DFS, Depth-limited search and Iterative deepening DFS. Heuristic Search Techniques: Hill Climbing, Best First Search, Problem Reduction, Constraint satisfaction. Means Ends Analysis.

UNIT -2 (12 Hours)

Knowledge Representation: Representation & Mapping, Approaches to knowledge Representation, Predicate Logic, Representing simple facts & logic. Representing instance & ISA relationship, Computable functions & predicates. Unification. Resolution. Symbolic Reasoning under uncertainly. Introduction to non-monotonic Reasoning. Logic for non-monotonic Reasoning. Weak slot and filter structures: Semantic nets, Frames. Strong Slot and Filter Structures: Conceptual dependency, Scripts

UNIT - 3 (12 Hours)

Game Playing: Mini Max Search Procedure, Adding alpha-beta cut offs. Planning: Components of a planning system, Blocks World Problem. Goal Stack Planning Non–linear Planning, Hierarchical Planning. Introduction to natural language processing:

UNIT - 4 (12 Hours)

Learning: Rote learning, learning by taking advice, learning in problem solving. Inductive learning: version space, decision trees: statistical learning: Naïve Bayes Technique. Neural networks: topology and activation functions, Perceptron learning algorithm and back propagation algorithm.
Expert systems: Characteristics and design, Types of expert system, Expert system shells

Recommended Readings:

1. Elaine Rich and Kevin Knight; Artificial Intelligence; Third Edition, TMH
2. Struart Russell and Peter Norvig ; Artificial Intelligence, a Modern Approach, Third Edition, Pearson Education.
3. Ela Kumar; Artificial Intelligence; I. K. International Publishing House, 2011.
4. Nils J. Nillson; Artificial Intelligence: A new Synthesis; Harcourt Asia
5. Patrick Winston; Artificial Intelligence; Pearson Education
6. George F. Luger; Artificial Intelligence : Structures and strategies for complex problem solving, Pearson education.
7. Saroj Kaushik; Artificial Intelligence; Cengage Learning, 2011

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Implementation of BFS
2. Implementation of Depth limited DFS
3. Implementation of Hill Climbing Search
4. Implementation of A* search
5. Implementation of Crypt arithmetic problem
6. Implementation of Unification Algorithm.
7. Implementation of TIC TAC TOE problem using Min-Max-Search
8. Implementation of an Expert System
9. Implementation of Logical functions using neural networks.

COMP 6.4 COMPUTER GRAPHICS

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 6.4	Computer Graphics	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

1. This course is designed to provide a comprehensive introduction to computer graphics leading to the ability to understand contemporary terminology, progress, issues, and trends.

2. A thorough introduction to computer graphics techniques, including 3D modeling, rendering and animation. Topics cover: geometric transformations, geometric algorithms, 3D object models (surface and volume), visible surface detection algorithms, image synthesis, shading and mapping, global illumination and animation techniques

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the concepts of computer graphics system
2. Implement the algorithms for two dimensional transformations.
3. Demonstrate the techniques of clipping
4. Explain the basics of 3D Graphics and three dimensional transformations.

UNIT - 1 (12 Hours)

Overview of graphic systems: Raster scans systems, Random scan systems.

Output Primitives. Points and lines, Line drawing algorithms, DDA, Bresenhams line algorithm Circle generating algorithms, Properties of circles, Midpoint circle algorithm, Ellipse generating algorithm, Properties of Ellipses, Midpoint ellipse algorithm, Filled area primitives, Scan line polygon Fill algorithm, Inside – outside tests, Scan line fill of curved boundary, Boundary fill algorithm, Flood fill algorithm, Fill area functions.

UNIT - 2 (12 Hours)

Two Dimensional Geometric Transformations: Basic Transformations, Translation, Rotation, Scaling, Composite transformation, Translations, Rotations, Scaling, Other transformations- Reflection, Shear.

Two-Dimensional Viewing: The viewing pipeline, Viewing coordinate reference frame, Window to viewport coordinate transformation, 2-D viewing functions, Clipping operations, Point Clipping, Line clipping , Cohen- Sutherland Line Clipping, Polygon Clipping, Sutherland Hodgeman Polygon clipping, Weiler- Atherton Polygon Clipping, Curve clipping, Text clipping.

UNIT - 3 (12 Hours)

Three Dimensional Concepts: 3- Dimensional display methods, Parallel projections Perspective projection, Depth cueing, Surface rendering, Exploded and cutaway views. Three Dimensional Object representations- Polygon surfaces, Polygon tables, Three Dimensional Geometric and Modeling transformations- Translation Rotation, Coordinate Axes, rotations , Scaling , Reflections , Shears
Three Dimensional Viewing, Transformation from world to viewing coordinates Projections.
Picture Structure: Defining Symbols By Procedures, Display Procedures, Boxing, Structured Display Files. Techniques for Achieving Realism. Curves And Surfaces: Shape Description Requirements, Parametric Functions, Bezier Methods. B-Spline Methods.

UNIT - 4 (12 Hours)

Classification of visible – surface detection algorithms, Back – Face detection , Depth buffer method , A – Buffer method , Scan – Line method , Depth Sorting method , BSP- Tree method, Area Sub-division method.

Color Models and Color Applications- Properties of light ,Standard primaries and the, Chromaticity Diagram, XYZ Color model, CIE Chromaticity Diagram, RGB color model, YIQ Color Model , CMY Color Model, HSV Color Model, HLS Color Model

Computer Animation: Design of animation sequences, General computer animation functions, Raster Animations, Computer animation languages, Motion specification, Direct motion specification, Goal directed systems Kinematics and dynamics.

Recommended Readings:

1. Donald Hearn and M. P. Baker ; Computer Graphics; Prentice Hall of India Pvt. Ltd.
2. William Newman and Robert Sproull; Principles of Interactive Graphics; Tata McGraw hill Publishing company Ltd.
3. N. Krishnamurthy; Introduction to Computer Graphics; TMH
4. Steven Harrington; Computer Graphics; Tata McGraw Hill.
5. Foley, Van Dam, Feiner and Hughe; Computer Graphics: Principles and Practice

List of Experiments in Computer Graphics:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. To study basic Graphics Primitive functions
2. To draw a Line using DDA line drawing algorithm
3. To draw a line using Breshenham's algorithm
4. To draw circle using midpoint circle algorithm
5. To draw an ellipse using mid-point ellipse algorithm.
6. To translate, rotate and scale the 2D object.
7. To translate, rotate and scale the 3D object.
8. To fill polygon using boundary fill algorithm.
9. To fill polygon using flood fill algorithm.
10. To implement Cohen–Sutherland 2D clipping and window–viewport mapping
11. To perform 2-D animation

COMP 6.5 EMBEDDED SYSTEM 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 6.5	Embedded System Design	3	1	2	3	100	25	-	-	25	150

Course Objectives:

The subject aims to provide the student with:

1. To conceptualize the basics of embedded systems.
2. To conceptualize the basics of organizational and architectural issues of a microcontroller
3. To learn programming techniques used in microcontroller
4. To understand fundamentals of real time operating system

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain about microcontrollers embedded processors and their applications
2. Develop the programs for microcontroller.
3. Describe the role of embedded systems in industry.

UNIT – 1 (12 Hours)

Overview of Embedded System Architecture, Application areas, Categories of embedded systems, specialties of embedded systems. Recent trends in embedded systems. Brief introduction to embedded microcontroller cores CISC, RISC, ARM, DSP and SoC (System on Chip). Introduction to 8051 Microcontroller, Architecture, Pin configuration, Memory organization, Input /Output Ports, Counter and Timers, Serial communication, Interrupts.

UNIT – 2 (12 Hours)

Assembly Language Programming of 8051: Instruction set, Addressing modes, Development tools, Assembler Directives, Programming based on Arithmetic & Logical operations, I/O parallel and serial ports, Timers & Counters, and Interrupt Service Routine.

UNIT – 3 (12 Hours)

Embedded / Real Time Operating System: Architecture of kernel, Task and Task scheduler, Interrupt service routines, Semaphores, Mutex, Mailboxes, Message queues, Event registers, Pipes, Signals, Timers, Memory management, Priority inversion problem. Off-the-Shelf Operating Systems, Embedded Operating Systems, Real Time Operating System (RTOS) and Handheld Operating Systems.

UNIT – 4 (12 Hours)

Embedded System - Design case studies: Digital clock, Battery operated smart card reader, Automated meter reading system, Washing Machine, Microwave Owen, Automotive Embedded Systems. Embedded software development tools: Code generation tools, Simulator, Testing and debugger, Integrated Development Environments (IDE) for 8051 systems, Memory and Processor sensitive program and device drivers.

Recommended Readings:

1. M. A. Mazidi, J. G. Mazidi, R. D. McKinlay; The 8051 microcontroller & Embedded systems; , Pearson
2. Kenneth J. Ayala, Dhananjay V. Gadre; . The 8051 microcontroller & Embedded systems; Cengage Learning.
3. Dr. K. V. K. K. Prasad; Embedded / real – time systems: concepts, design & programming, Black Book; Dreamtech press, Reprint edition 2013.
4. Raj Kamal; Embedded System: architecture, programming and design; TMH.
5. Frank Vahid; Tony Givargis,;John Wiley; Embedded System Design;
6. Laya B. Das, Pearson; Embedded systems an integrated approach;
7. Frank Vahid,;Tony Givargis;Embedded system design A Unified hardware/software Introduction.
8. Shibu K.V; Introduction to Embedded Systems; Mc Graw Hill

List of Experiments:

(At least 8 experiments should be conducted based on the broad areas listed below)

Using Keil

1. Write a program to send ASCII values 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e to port 1
2. Write a program to toggle the bits of P1
3. Write a program to send and receive data serially
4. Programming based on arithmetic operations in 8051
5. Programming based on logical operations in 8051
6. Programming based on timers in 8051
7. Programming based on interrupts in 8051

Based on RTOS

1. To implement Shortest Job First Scheduling algorithm
2. To implement Priority Inheritance Protocol
3. Case Study: Reliability & Fault tolerance in RTOS

Case Study on Embedded System

1. Digital clock,
2. Battery operated smart card reader,
3. Automated meter reading system,
4. Washing Machine,
5. Microwave Owen,
6. Automotive Embedded Systems

COMP 6.6 COMPUTER NETWORKS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	ThDuration (Hrs)	Marks					
						Th	S	TW	P	O	Total
COMP 6.6	Computer Networks	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. To provide an introduction to basic concepts of communication and networks.
2. To understand the underlying principles and approaches in computer networks.
3. To provide students knowledge of data link, network, transport and application protocols.
4. To equip students with necessary skills required for network programming.

Course Outcomes:

The student after undergoing this course will be able to:

1. Understand the fundamental concepts of computer networks, its evolution and architectures.
2. Have a good understanding of the OSI and TCP/IP Reference models
3. Have a detailed understanding of data link, network, transport and application layer protocols.
4. Analyze the requirements for a given organization and select the most appropriate networking architecture and technologies.
5. Have knowledge of network programming and tools.

UNIT -1 (12 Hours)

Reference Models: The OSI Reference Model, The TCP/IP Reference Model, A Comparison of the OSI and TCP/IP Reference Models. The Physical Layer: The Theoretical Basis for Data Communication, Fourier Analysis, Bandwidth-Limited Signals, The Maximum Data Rate of a Channel. Data Link Layer Design Issues, Error-Correcting Codes, Error –Detecting Codes, Elementary Data Link Protocols- An Unrestricted Simplex Protocol, A Simplex Stop-and-Wait Protocol, A Simplex Protocol for a Noisy Channel, Sliding Window Protocols - A One-Bit Sliding Window Protocol, A Protocol Using Go Back N.

UNIT - 2 (12 Hours)

Multiple Access Protocols – ALOHA, Carrier Sense Multiple Access Protocols, Collision-Free Protocols, Limited-Contention Protocols, Wavelength Division Multiple Access Protocols, Wireless LAN Protocols, Ethernet - Ethernet Cabling, Manchester Encoding, The Ethernet MAC Sublayer Protocol. Network Layer Design Issues, Routing Algorithms- Shortest Path Routing, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Broadcast Routing, Multicast Routing, Congestion Control Algorithms -General Principles of Congestion Control, Congestion Prevention Policies, Congestion Control in Virtual-Circuit Subnets and Datagram Subnets, Load Shedding. The IP Protocol, IP Addresses, Internet Control Protocols.

UNIT - 3 (12 Hours)

The Transport Layer: The Transport Service, Services Provided to the Upper Layers, Transport Service Primitive, Berkeley Sockets. Elements Of Transport Protocols-Addressing, Establishing a Connection, Releasing a Connection, The Internet Transport Protocols: UDP, TCP- The Service Model, The protocol, The TCP Segment Header, TCP Connection Establishment, TCP Connection Release. The Application Layer: Domain Name System- Name Space, Resource Records, Name Servers, Electronic Mail- Architecture and Services, The User Agent, Message Formats, Message Transfer, Final Delivery

UNIT - 4 (12 Hours)

Introduction to NS2: Simulator structure, Simulator input and Output, NS2 Installation steps, NS2 Directories & Files, Network Animator, NS2 Program Structure. Basics of Protocol Simulation Using NS2: Tcl, Program execution, Basic programming constructs, arrays, Lists, Dictionaries, Procedures, File handling, Object oriented Tcl, General structure of AWK Scripting, Gnuplot. Wired Network simulation and wireless network simulation.

Recommended Readings:

1. Andrew S. Tannenbaum; Computer Networks; PHI, 4th Edition
2. Ajit Kumar Nayak, Satyananda Champati Rai, Rajib Mall; Computer Network Simulation Using NS2,CRC.
3. Behrouz A. Forouzan; Data communication and Networking; Tata McGraw Hill.
4. Williams Stallings; Data and Computer Communications; PHI.
5. Fred Halsall; Data Communications, Computer Networks and Open Systems; Pearson Education

List of Experiments in Computer Networks:

(At least 8 experiments should be conducted from the list of experiments. A certified journal reporting the experiments conducted should be submitted at the end of the term)

1. Study of Networking and Internetworking devices
2. Implementation error detection mechanisms
3. Implementation of error correction codes
4. Study of Network Layer- IP Addresses
5. Implementation of shortest Path Routing algorithms
6. Socket Programming in TCP/UDP
7. Installation of NS2
8. Implementation of Destination sequenced distance vector
9. Implementation of Dynamic source routing
10. Implementation of Temporally ordered routing algorithm
11. Implementation of Adhoc on demand distance vector

APPENDIX

QUESTION PAPER PATTERN

Syllabus in each subject will have 4 units.

Question paper shall be drawn as follows:

Question No	From Units	No. of Questions to be Set	No. of Questions to be Answered	Remarks
1-3	1-2	3 x 20marks	2 x 20 marks	Each unit shall have minimum 20 marks
4-6	3-4	3 x 20 marks	2 x 20 marks	Each unit shall have minimum 20 marks
7-8	1-4	2 x 20 marks	1 x 20 marks	---
		8 - 160 marks	5 - 100 marks	

SAMPLE QUESTION PAPER

SUBJECT: MARKS: 100

MAXIMUM DURATION: 3 hours

Instructions to the candidates:

1.

2

Part –A (Questions to be drawn from units 1 & 2)

Answer any **TWO** questions from the following: 2 x 20= 40 Marks

Question-1 ....20 Marks

a)

b)

..

Question-2 ....20 Marks

a)

b)

..

Question-3 ....20 Marks

a)

b)

..

Part –B (Questions to be drawn from units 3 & 4)

Answer any **TWO** questions from the following: 2 x 20= 40 Marks

Question-4 ....20 Marks

a)

b)

..

Question-5 ....20 Marks

a)

b)

..

Question-6 ....20 Marks

a)

b)

..

Part –C (Questions to be drawn from all units i.e. units 1 - 4)

Answer any **ONE** question from the following: 1 x 20= 20 Marks

Question-7 ....20 Marks

a)

b)

..

Question-8 ....20 Marks

a)

b)

..

..